



DIPLOMARBEIT

Herr
Martin Röhlig

**Visualisierung und intuitive
Interaktion für komplexe
Produktstrukturen**

2011

DIPLOMARBEIT

Visualisierung und intuitive Interaktion für komplexe Produktstrukturen

Autor:

Martin Röhlig

Studiengang:

Multimediatechnik

Seminargruppe:

MK06W2

Erstprüfer:

Prof. Dr. rer. nat. Günter Werner

Zweitprüfer:

Dipl.-Ing. Des. Ernst-Eckart Schulze

Mittweida, Januar 2011

Bibliografische Angaben

Röhlig, Martin: Visualisierung und intuitive Interaktion für komplexe Produktstrukturen, 135 Seiten, 60 Abbildungen, Hochschule Mittweida (FH), Fakultät Informationstechnik und Elektrotechnik

Diplomarbeit, 2011

Dieses Werk ist urheberrechtlich geschützt.

Referat

Die Diplomarbeit beschäftigt sich mit der Entwicklung eines grafischen Managementsystems für komplexe Produktstrukturen.

Effektive Visualisierungs- und intuitive Interaktionskonzepte für und mit den Strukturen werden untersucht und in das System integriert. Weitere Schwerpunkte sind die Erforschung und Analyse entsprechender Technologien und die Modularität, Schnittstellengestaltung und Implementation des Systems auf einer Multi-Touch-Konsole.

Ziel ist es, durch neuartige Visualisierungs- und Interaktionstechniken die Arbeit mit Produktstrukturen effektiver zu gestalten und komplexe Datensysteme versteh- und beherrschbar zu machen. Mit einer Benutzerstudie werden die erzielten Ergebnisse analysiert, diskutiert und bewertet.

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	II
Tabellenverzeichnis	III
Abkürzungsverzeichnis	IV
Vorwort	V
1 Einleitung	1
1.1 Ausgangssituation	1
1.2 Zielsetzung	3
1.3 Vorgehensweise	4
2 Grundlagen	7
2.1 Definition von Produkt	7
2.2 Produktstrukturen und elementare Ausprägungen	8
2.2.1 Aggregation	9
2.2.2 Assoziation	10
2.2.3 Variation	10
2.2.4 Veränderung	11
2.3 Produktrepräsentation und -präsentation	12
2.4 Visualisierung und Data Mining	14
2.4.1 Visualisierung	14
2.4.2 Visualisierungspipeline	21
2.4.3 Data Mining	23
2.5 Interaktion und Navigation	27
2.6 Klassifikation von Visualisierungs- und Interaktionskonzepten	32
2.7 Benutzungsschnittstellen und Präsentation	38
3 Stand der Technik	45
3.1 Produktstrukturen innerhalb von Anwendungssystemen	45
3.1.1 CAD-Systeme	45
3.1.2 PDM-Systeme	47
4 Anforderungen und Handlungsbedarf	51
4.1 Eignungskriterien für Visualisierungssysteme	51
4.2 Integrationsaspekte	53
4.3 Erstellungsaufwand und Nutzen	55
4.4 Handlungsbedarf	56
5 Entscheidungsfindung und Konzeption	59
5.1 Entscheidungsfindung	59
5.1.1 Auswahl der Visualisierungskonzepte	59

5.2	Konzeption des Visualisierungs- und Interaktionssystems (VIS)	63
6	Systemtechnische Realisierung	67
6.1	Realisierung der Integrationsschnittstelle	68
6.2	Realisierung der Visualisierungskonzepte	71
6.2.1	Implementierung des Kapselung-Layouts	72
6.2.2	Implementierung des Knoten-Kanten-Layouts	77
6.3	Realisierung der Datenhaltung	82
6.4	Realisierung der Benutzungsschnittstelle und Interaktionsgestaltung	82
6.5	Erweiterungen mittels Multi-Touch Technologie	89
7	Evaluation und Validierung	93
7.1	Hypothesen zur Benutzerstudie	94
7.2	Testverfahren und -ablauf	95
7.3	Auswertung der Ergebnisdaten	102
7.4	Diskussion und weiterführende Betrachtung	112
8	Zusammenfassung und Ausblick	115
	Literaturverzeichnis	117
A	Exemplarischer Fragebogen der Benutzerstudie	125
B	Exemplarische Generalisierung des VIS	133

Abbildungsverzeichnis

1.1	Komplexität in der Produktentwicklung	2
1.2	Vorgehensweise in der Arbeit	5
2.1	Elemente und Beziehungen innerhalb von Systemen	8
2.2	Aggregation in Bezug auf Produktstrukturen	9
2.3	Assoziation in Bezug auf Produktstrukturen	10
2.4	Variation in Bezug auf Produktstrukturen	11
2.5	Veränderung in Bezug auf Produktstrukturen	12
2.6	Repräsentation und Präsentation von Produktdaten	13
2.7	Bäume als Produktstrukturpräsentation	14
2.8	Klassisches Beispiel für die Visualisierung	16
2.9	Kluft der Ausführung und Evaluation	19
2.10	Ein Bezugssystem für die Taxonomie der Informationsvisualisierung	20
2.11	Referenzmodell der Visualisierungspipeline	21
2.12	Architektur eines Data Mining Systems	25
2.13	Prozess des Data Mining	26
2.14	Interaktionsarten der Informationsvisualisierung	31
2.15	Beispiele für die Visualisierungskategorie Kapselung	34
2.16	Beispiele für die Visualisierungskategorie Knoten-Kanten	36
2.17	Beispiele für Relationen zwischen Visualisierungen	37
2.18	Beispiele für technisch-wissenschaftliche Visualisierung	38
2.19	Beispiele für Bildlauf, Overview plus Detail und ZUI Interfaces	40
2.20	Beispiele für Fokus und Detail-In-Kontext Techniken	41
2.21	Beispiele für diverse NUI und Präsentationsmedien	43
2.22	Beispiele für Multi-Touch Technologien	44
3.1	Visualisierung innerhalb von CAD-Systemen	46
3.2	Visualisierung innerhalb von PDM-Systemen	48
4.1	Auswahlkriterien und Anforderungen an Visualisierungen	52
5.1	Slice-And-Dice und Squarified Treemap-Layouts	60

5.2	Radiale Knoten-Kanten-Layouts	62
5.3	Konzeption der Gebrauchstauglichkeit	64
5.4	Konzeption der Systemarchitektur	65
6.1	Architektur der Integrationsschnittstelle	71
6.2	Prinzipielle Datenabbildung mittels Treemap-Layouts	72
6.3	Beispiele für unbegrenzte und begrenzte Voronoi Tessellation	73
6.4	Beispiele für additiv gewichtete Voronoi Tessellation	74
6.5	Beispiele zentrischer Voronoi Tessellation (CVT)	75
6.6	Beispiele für Voronoi-Treemaps	76
6.7	Darstellungsvarianten von Voronoi-Treemaps	78
6.8	Generierungsprozess des Knoten-Kanten-Layouts	81
6.9	Darstellungsvarianten von Force-Directed-Layouts	81
6.10	Konzept der Benutzungsschnittstelle	84
6.11	Explorationsmodus der Benutzungsschnittstelle	85
6.12	Automatisierung und Animationen der Benutzungsschnittstelle	87
6.13	Grafische Benutzungsoberfläche und Bedienelemente	88
6.14	Interaktionstechniken mit Multi-Touch-Gesten	90
7.1	Einführende Visualisierungslayouts der Benutzerstudie	97
7.2	Erster einleitender Strukturtest der Benutzerstudie	98
7.3	Zweiter einleitender Strukturtest der Benutzerstudie	98
7.4	Visualisierungslayouts für die Aufgabenbearbeitung der Benutzerstudie	100
7.5	Auswertung der Vorkenntnisse der Probanden	103
7.6	Auswertung der Bearbeitungszeiten des 2. Strukturtests	104
7.7	Auswertung der Fehlerraten der Aufgabenstellungen	105
7.8	Fehlerquellen in der Aufgabenbearbeitung	106
7.9	Auswertung des Schwierigkeitsgrades	107
7.10	Auswertung der Gesamtbearbeitungszeit	108
7.11	Auswertung der Aufgabenbearbeitungszeiten	108
7.12	Auswertung der Interaktionen	109
7.13	Auswertung der Einschätzungen	111
7.14	Auswertung der Layout-Bevorzugung	112

B.1	Visualisierung eines Softwaresystems	133
-----	--	-----

Tabellenverzeichnis

2.1 Metadaten-Klassen für Produktdaten	7
2.2 Relevanz der Visualisierung	15
2.3 Merkmale der Visualisierungskategorie Kapselung	34
2.4 Merkmale der Visualisierungskategorie Knoten-Kanten	35
4.1 Positive Aussagen zu bestehenden PDM-Systemen	57
4.2 Negative Aussagen zu bestehenden PDM-Systemen	57
5.1 Eigenschaften von Treemap-Algorithmen	61
6.1 Allgemeiner Aufbau der Datenhaltung	83
7.1 Ausschnitt einer Logdatei der Benutzerstudie	96

Abkürzungsverzeichnis

AR	Augmented Reality
B-Rep	Boundary Representation
CAD	Computer-Aided Design
CAVE	Cave Automatic Virtual Environment
CLI	Command-Line Interface
CSG	Constructive Solids Geometry
DBMS	Datenbankmanagementsysteme
DTD	Document Type Definition
DTP	Desktop Publishing
EB	Exabyte
ERM	Entity-Relationship-Modell
FEM	Finite Element Method
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HCI	Human Computer Interaction
IxD	Interaction Design
KDD	Knowledge Discovery in Databases
LoD	Level-Of-Detail
MB	Megabyte
MRO	Maintenance, Repair and Overhaul
MT4J	Multitouch for Java
MVC	Model-View-Controller
NUI	Natural User Interfaces
O+d	Overview plus Detail Interface
OLAP	Online Analytical Processing
PDM	Produktdatenmanagement
PLMXML	eXtensible Markup Language for Product Lifecycle Management
PPS	Produktionsplanungs- und Steuerungssystem
STEP	STandard for the Exchange of Product model data
UML	Unified Modeling Language
VIS	Visualisierungs- und Interaktionssystem
VR	Virtual Reality

W3C	World Wide Web Consortium
WIMP	Windows, Icons, Menus, Pointer
XML	eXtensible Markup Language
XSD	XML-Schema-Definition
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Language Transformations
ZUI	Zoomable User Interface

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als Diplomand am Fraunhofer Institut für Produktionsanlagen und Konstruktionstechnik in der Abteilung Virtuelle Produktentstehung Berlin, im Zeitraum von April 2010 bis Januar 2011.

Mein besonderer Dank gilt meinem Betreuer Herrn Dipl.-Ing. Des. Ernst-Eckart Schulze für die gute Zusammenarbeit, die kritische Auseinandersetzung mit meiner Arbeit und die hilfreichen Diskussionsrunden. Bei Herrn Prof. Dr. rer. Nat. Günter Werner bedanke ich mich vornehmlich für seine Unterstützung, das Interesse und fortwährende Vertrauen in meine Arbeit. Für die technischen Denkanstöße und Hilfestellungen möchte ich vor allem Herrn Prof. Dr. Oliver Deussen und Dipl.-Inform. Lars Wolter danken sowie allen Mitarbeitern der Abteilung Virtuelle Produktentstehung für die gute Atmosphäre und das nette Miteinander während und neben der Arbeit.

Mein herzlichster Dank geht darüber hinaus an meine Familie und Freunde für die vielfältige Unterstützung vor, während und fernab von der Arbeit. Wiot!

1 Einleitung

1.1 Ausgangssituation

Die Produktentwicklung ist seit einiger Zeit mit einer steigenden Komplexität konfrontiert, welche vor allem aus einer anhaltenden Technisierung und kundenspezifischen Anpassungen von Produkten resultiert. Weitere Ursachen sind die hochwertige Funktionalität und der Variantenreichtum heutiger Produkte, um eine Abgrenzung zur Konkurrenz auf den wettbewerbsintensiven Märkten zu ermöglichen. Auswirkungen hat diese Entwicklung auf den gesamten Produktlebenszyklus, wobei Produktentwicklungsprozess und -management besonders betroffen sind. So müssen auch mit zunehmender Komponentenanzahl und erhöhtem Administrations- und Koordinationsaufwand die Qualität stets gewährleistet und Entwicklungszeiten eingehalten werden [KFG06].

Organisatorische Methoden und technologische Ansätze sollen bei der Komplexitätsbewältigung helfen. Besonders der Einsatz von informationstechnischen Werkzeugen, wie Produktdatenmanagementsysteme (PDM-Systeme) und Internettechnologien fördern Transparenz, Verkürzung von Entwicklungszeiten und Organisationssupport bei verteilter Produktentwicklung. Wesentlicher Bestandteil dieser Ansatzpunkte bilden Strukturen, die bei der Produktentwicklung entstehen und Informationen zum gesamten Produktlebenszyklus beinhalten. Diese Produktstrukturen sind komplexe Datenstrukturen, welche in verschiedenen Ausprägungen zur Komplexitätsreduktion, höheren Übersichtlichkeit, Nachvollziehbarkeit und für verbesserte Exploration eingesetzt werden. Es handelt sich dabei um hierarchische oder netzartige Ordnungssysteme, welche Produkte und deren Einzelkomponenten in logischem Zusammenhang beschreiben.

Heutige Produktstrukturen enthalten neben einer hierarchischen Ordnung auch verschiedene Dimensionen der Komplexität, wie spezifische Einzelteil-Versionen, Varianten-Optionen und spezielle Beziehungen der einzelnen Strukturknoten. Somit sind bei hochtechnischen Produkten und durch die Verschmelzung von Hardware und Software, der Kontext und die spezifischen Attribute einzelner Elemente und Baugruppen kaum noch beherrschbar bzw. wahrnehmbar. Weitere Defizite liegen in der nicht standardisierten Präsentation und Generierung und bei der starken Trennung von 3D-Geometriemodellen und Produktstrukturen. Hohe Abstraktionsfähigkeit und Arbeitsbelastung der Produktentwickler oder sogar Interpretationsprobleme und daraus resultierende Fehler sind oftmals die Folge.

Im Bereich der Informationsvisualisierung gibt es bereits seit einiger Zeit Bestrebungen und Ansätze, um große Informationsmengen und deren Komplexität darzustellen [War04, Che04, Maz09]. Vorteil solcher Visualisierungskonzepte sind eine verbesserte kognitive Wahrnehmung und leichteres Verständnis von Kontext und Verhältnis. Oftmals werden

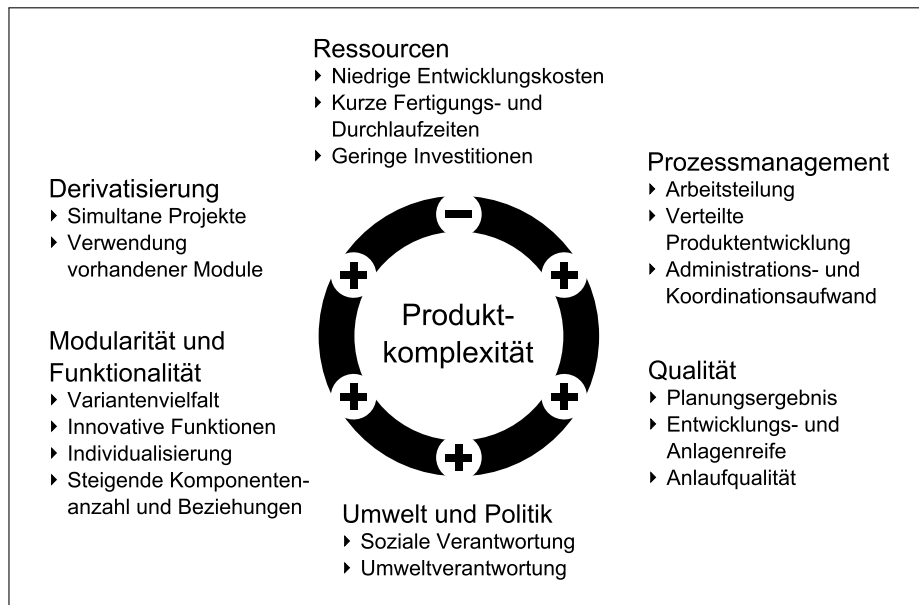


Abbildung 1.1: Ursachen und Auswirkungen von Komplexität in der Produktentwicklung angelehnt an [Sei08]

implizite Informationen explizit sichtbar, welche aus dem bloßen Informationsgehalt der Struktur nicht erfassbar sind.

Derartige Methoden werden momentan in produktspezifischen Anwendungen allerdings nur bedingt integriert und kommen in konkreten Projekten kaum zur Umsetzung. Zugleich haben derzeit zum Einsatz kommende Visualisierungskonzepte meist einen statischen Charakter. Daher ist es Nutzern nicht oder nur über ineffiziente Umwege möglich, mit den dargestellten Informationen zu interagieren und zu navigieren.

Gleichzeitig bieten aktuelle Präsentationsmedien neuartige und alternative Benutzungsschnittstellen und erlauben einen stark erhöhten Interaktionsgrad. In den Vordergrund rücken das Erleben der Anwendung und der intuitive Umgang mit dem eigentlichen Informationsgehalt. Gegenwärtige Entwicklungen befassen sich neben traditionellen Eingabegeräten, wie Maus und Tastatur, auch mit alternativen und natürlichen Interaktionsansätzen. Der Einsatz herkömmlicher Objekte und die Analyse von Gesten oder Berührungen mit den Fingerspitzen führen zu neuen Manipulations- und Interaktionsmodellen. Ferner wird es durch die Unterstützung simultaner Eingabegeräte und -punkte möglich, dass mehrere Benutzer gleichzeitig mit einer Anwendung interagieren und kommunizieren.

Hauptschwerpunkte sind die Auswahl, Darstellung und Integration von zweckdienlichen und dem Kontext entsprechenden Konzepten. Zur effektiven Nutzung des verbundenen Optimierungspotenzials ist ferner die Anpassung der Modelle an die spezifischen Anforderungen unterschiedlicher Applikationen unerlässlich.

1.2 Zielsetzung

Die Motivation der Arbeit besteht darin, ein flexibles Managementsystem für existierende Produktstrukturen zu schaffen. Mögliche Verbesserungsmaßnahmen und Optimierungsansätze für derzeitige Präsentationstechniken werden anhand neuer Visualisierungs- und Interaktionstechniken diskutiert und prototypisch umgesetzt. Besondere Schwerpunkte sind die Bezugnahme auf die spezifischen Anforderungen der Produktentwickler, die Berücksichtigung der Voraussetzungen von Produktstrukturen und die Analyse und Auswertung der Ergebnisse.

Es werden unter anderem Funktionen für Import und Export, benutzerfreundliche Darstellung und intuitive Navigation durch und mit Produktstrukturen implementiert. Bei der systemtechnischen Realisierung wird besonderer Wert auf die Modularität und Systemunabhängigkeit gelegt. Zur Förderung der Nutzerakzeptanz findet darüber hinaus die Anbindung an bestehende Systeme Berücksichtigung.

Die Arbeit widmet sich somit der Entwicklung einer dynamischen Anwendung zur einfachen Handhabung der Datenmenge und effizienten Umgang mit Produktstrukturen. Insbesondere werden geeignete Darstellungskonzepte integriert, um den Informationsgehalt wahrnehmbar zu machen. Dabei wird untersucht, welche Visualisierungs- und Interaktionskonzepte sich für die Darstellung der Strukturen bzw. deren Navigation eignen. Einen weiteren wichtigen Bestandteil bildet die Validierung der Expressivität, Effektivität und Angemessenheit der entwickelten Konzepte bezüglich der Zielgruppe und der Einsatzfelder. Durch Nutzung der Präsentations- und Interaktionsmöglichkeiten wird der statische Informationsspeicher – die Produktstruktur – in eine dynamische, beeinflussbare Form überführt. Anliegen ist es, Produktentwickler, -designer und -manager bei der Arbeit, Planung und Kollaboration zu unterstützen. Zum Beispiel könnten benötigte Informationen zu einem bestimmten Produktentwicklungsstand gezielt abgefragt oder anhand spezifischer Produktattribute nach genauen Vorgaben erstellt werden. Durch die Verwendung des Systems sind einzelne Produktentstehungsabschnitte leicht nachvollziehbar und es können Festlegungen über die weitere Konzeption der Komponenten getroffen werden. Weiterhin wird es möglich, eine spezifische Version des Produkts aus der Gesamtstruktur und nach speziellen Kundenanforderungen zu definieren.

Als Erweiterung wird eine konkrete Anpassung des Systems für den Einsatz auf Multi-Touch-Geräten geschaffen. Die adaptierte Funktionalität der Benutzungsschnittstelle umfasst unter anderem Gestenunterstützung, Multi-User Management und simultane Eingabemethoden. Die Möglichkeiten des Mediums werden dabei effektiv genutzt, um kollaborative Arbeitsprozesse weiter zu fördern.

1.3 Vorgehensweise

Zur Verwirklichung der gesetzten Zielstellung folgt die Arbeit einer systematischen Vorgehensweise. Nach diesem einleitenden **Kapitel 1** gliedert sich die Arbeit in folgende sieben inhaltliche Bestandteile:

Kapitel 2 stellt die theoretischen Grundlagen zu Produktstrukturen und Visualisierungs- und Interaktionstechniken vor. Weiterhin werden die Potenziale innovativer Präsentationsmedien und natürlicher Benutzungsschnittstellen beschrieben und eine generelle Klassifikation von Visualisierungs- und Interaktionskonzepten vorgenommen.

Kapitel 3 analysiert den Stand der Technik im Umfeld der Visualisierung und Interaktion von Produktstrukturen. Der Schwerpunkt liegt auf den spezifischen Präsentationsformen innerhalb von erzeugenden und prozessbegleitenden Anwendungssystemen.

Kapitel 4 definiert die generellen Anforderungen an ein gebrauchstaugliches Visualisierungssystem. Die wesentlichen Eignungskriterien, die anwendungskontextspezifischen Integrationsaspekte, der Einbezug des Erstellungsaufwands und der daraus resultierende Nutzen für den Anwender werden diskutiert. Aus den gewonnenen Richtlinien wird folglich der konkrete Handlungsbedarf für die Arbeit abgeleitet.

Kapitel 5 bestimmt die relevanten Entscheidungen der Arbeit und gibt eine zusammenfassende Übersicht über die informationstechnische Konzeption zur Umsetzung eines Visualisierungs- und Interaktionssystems für Produktstrukturen.

Kapitel 6 widmet sich der systemtechnischen Realisierung des Visualisierungs- und Interaktionssystems auf der Grundlage der Anforderungen. In diesem Zusammenhang wird die prototypische Entwicklung der einzelnen Systemkomponenten beschrieben. Die Implementierung von zwei Visualisierungskonzepten für die Aggregationsabbildung von Produktstrukturen wird dabei eingehend dargestellt. Die Entwicklung eines Demonstrators und dessen Adaption an natürliche Benutzungsschnittstellen bilden den Abschluss des Kapitels.

Kapitel 7 umfasst das Verfahren und die Auswertung einer Benutzerstudie zur Softwarequalitätssicherung des entwickelten Systems. Die Validierung der einzelnen Softwarekomponenten und die Angemessenheit bezüglich Einsatzfeld und Zielgruppe werden vorrangig analysiert. Die Formulierung und Konkretisierung von geeignetem Optimierungspotenzial ist ein weiterer wichtiger Bestandteil der Auswertung.

Kapitel 8 fasst die wesentlichen Ergebnisse der vorliegenden Arbeit zusammen. Ein abschließender Ausblick verdeutlicht den zukünftigen Forschungsbedarf im Bereich der Visualisierung und Interaktion mit Produktstrukturen und zeigt weitere potenzielle Anwendungskontexte auf.

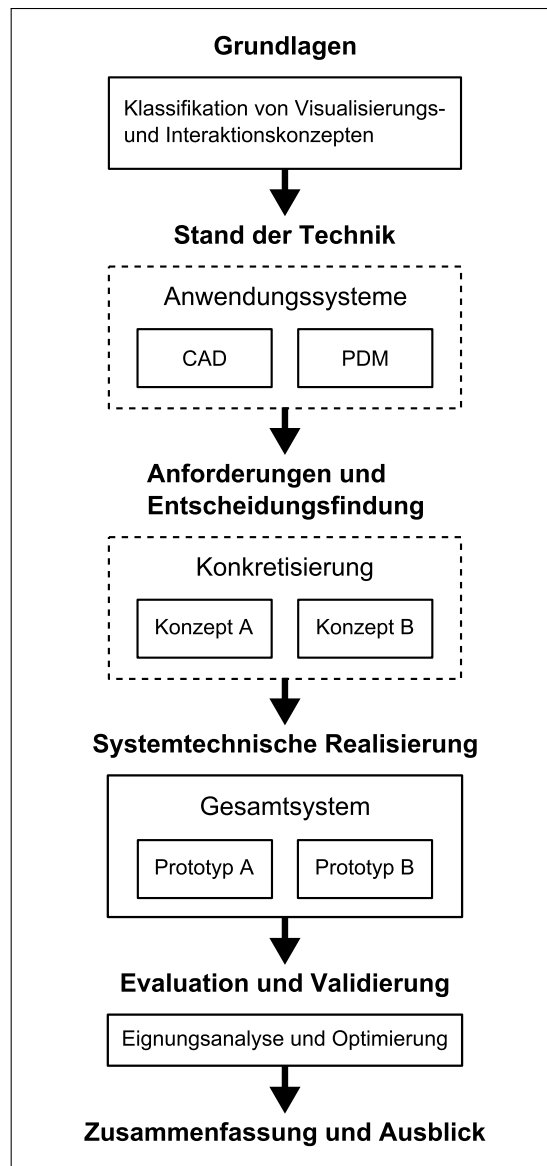


Abbildung 1.2: Schematische Darstellung der Vorgehensweise in der Arbeit

2 Grundlagen

2.1 Definition von Produkt

Im Rahmen dieser Arbeit wird der Begriff **Produkt** als umgangssprachliches Synonym für Erzeugnis oder Stückgut verwendet. Dies spiegelt die klassische Formulierung aus der Volkswirtschaftslehre wieder, wobei überwiegend auf industriell gefertigte, physische Objekte Bezug genommen wird, nicht aber auf die Bezeichnung einer Dienstleistung als Produkt. Dementsprechend ist der Begriff wie folgt definiert [DIN03]:

„Erzeugnisse sind in sich geschlossene, aus einer Anzahl von Gruppen und/oder Teilen bestehende funktionsfähige Gegenstände (z. B. Maschinen, Geräte) als Fertigungs-Endergebnisse.“

In der Produktentwicklung und -konzeption werden Produkte anhand von verschiedenen **Datenmodellen** beschrieben, welche es ermöglichen, eine Analogie zu den späteren realen Objekten herzustellen. Als zentrales Element der Abstraktion werden hierbei **Metadaten**¹ eingesetzt, die ein effizientes Management von Produktdaten² ermöglichen. Wie in Tabelle 2.1 dargestellt, lassen sich verschiedene Klassen unterscheiden.

Metadaten	Beispiele
Identifizierend	Bezeichnung (Name, Nummer), Ersteller (Unternehmen), Projektzugehörigkeit, Erstellungsdatum, Variantenummer, Änderungsstand, Versionsnummer
Klassifizierend	Erzeugersystem, Systemversion, Dateiformat (.model, .jpg, .vrm, .doc)
Organisatorisch	Ablageort (URL), Zugriffsrechte, technologische Voraussetzungen, Kosten, Status
Erweitert	Textuelle Beschreibung, Ansichteninformationen, Schnellansicht

Tabelle 2.1: Metadaten-Klassen und Beispiele nach [Les01]

Metadaten enthalten unter anderem Nummerierungen, mit denen Produktdaten eindeutig identifiziert und klassifiziert werden können. Diese Ziffern- und Zahlenkodierung ist insbesondere bei der Arbeit mit Produktstrukturen für Übersicht und Suche nach Produktdaten von Vorteil.

¹ Metadaten sind allgemeine Daten, die zumeist beschreibende Informationen zu anderen Daten enthalten.

² Zu Produktdaten gehören produktdefinierende, -repräsentierende, -präsentierende Daten und Dokumente, welche in Folge der Produktentwicklung entstehen.

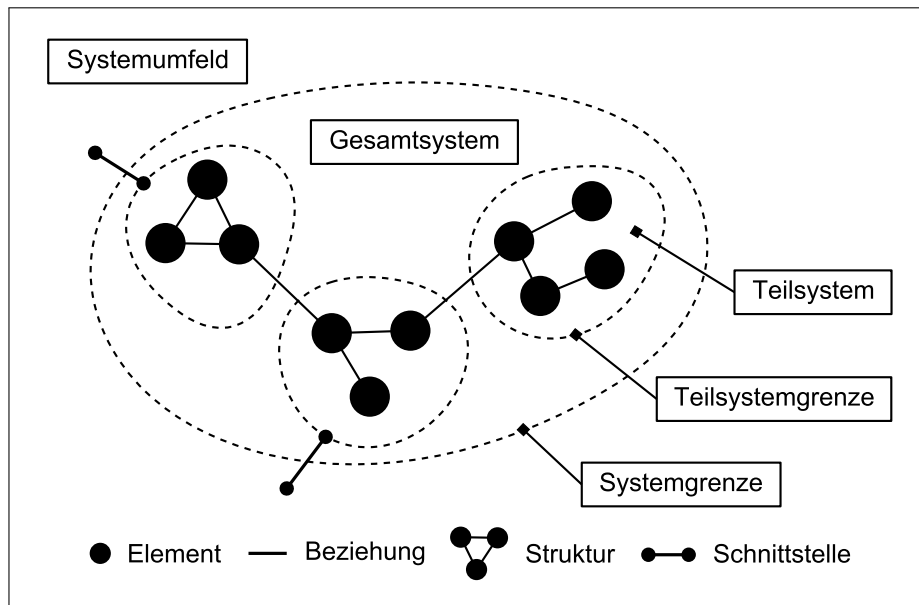


Abbildung 2.1: Elemente und deren Beziehungen innerhalb von Systemen nach [Les01, HD02]

2.2 Produktstrukturen und elementare Ausprägungen

Produktstrukturen sind dem Begriff nach **Strukturen**, welche wie folgt definiert werden [Wik10]:

„Unter Struktur versteht man das Muster von Systemelementen und ihrer Wirkbeziehungen (Relationen) untereinander, also die Art und Weise, wie die Elemente eines Systems aufeinander bezogen sind (durch Beziehungen „verbunden“ sind), so dass ein System bzw. Organismus funktioniert.“

Strukturen beschreiben also die Wechselwirkung von Systemelementen in einem **System**. Im Bereich der Produktentwicklung sind **Elemente** meist Abbildungen von physikalischen Gegenständen oder abstrakte Sachverhalte und besitzen dementsprechend eine individuelle Identität, definiertes Verhalten und Attribute. Dazu gehören vor allem Bauteile eines zu entwickelnden Produkts, Wirkprinzipien, Teilfunktionen und Anforderungen.

Zur Komplexitätsreduktion werden gleichartige Elemente anhand deren Charakteristik zu **Klassen** zusammengefasst und weiter in Ober- und Untergruppen gegliedert. Dadurch entsteht eine hierarchische Einteilung zwischen den Elementen und Klassen, was die Arbeit mit sehr großen und facettenreichen Informationsmengen stark vereinfacht. Generell wird dabei davon ausgegangen, dass Klassen mit großer Hierarchietiefe spezielle und Klassen geringer Tiefe allgemeine Elemente beinhalten. Weiterhin können Systeme aus einer Menge von Teilsystemen und atomaren Elementen³ bestehen. Diese werden durch das System von der Umwelt abgegrenzt und bilden durch deren **Beziehungen** das Systemverhalten.

³ Ein atomares Element ist nicht weiter in seine Bestandteile zerlegbar.

Produktstrukturen gehören zu den Strukturarten von **technischen Systemen**, wobei die Systemelemente durch Bauteile und -gruppen repräsentiert werden. Diese besitzen spezifische Attribute (z.B. Größe, Gewicht, Material) und Verhaltensweisen (z.B. kinetische Modelle) und sind durch verschiedene Beziehungen miteinander verbunden. Für den Begriff der Produktstruktur werden unterschiedliche Synonyme (z.B. Strukturstückliste, Baustruktur, Erzeugnisbaum) verwendet, welche aber im Kern, Beschreibung von Beziehungen von Bauteilen und Baugruppen eines Produkts, semantisch gleichbedeutend sind. Die Bezeichnung kann somit nach [DIN77] wie folgt abgegrenzt werden:

„Die Produktstruktur ist ein produktdarstellendes Modell, dass die Gesamtheit der nach bestimmten Gesichtspunkten festgelegten Beziehungen zwischen Baugruppen und Einzelteilen eines Produktes beschreibt.“

Des Weiteren lassen sich zu Produktstrukturen vier **elementare Ausprägungen** [Les01] identifizieren. Dazu gehören Aggregation, Assoziation, Variation und Veränderung.

2.2.1 Aggregation

Der hier verwendete Begriff **Aggregation** ist dem Entity-Relationship-Modell (ERM) [Che76, SS77] entnommen und bezieht sich auf die hierarchischen Zusammenhänge zwischen Einzelelementen und zu Klassen zusammengefassten Elementen. Der Einsatz von ERM ist der De-facto-Standard für die Datenmodellierung und dient dazu, meist anhand einer Grafik (z.B. Unified Modeling Language (UML) Diagramme), einen Ausschnitt der realen Welt zu beschreiben. Dabei können Klassen auch als Aggregate und deren Einzelelemente als Komponenten bezeichnet werden. In Verbindung mit Produktdaten sind jene Elemente und Klassen mit is-part-of-Beziehungen untereinander verbunden und bilden somit einen gerichteten Graphen. So darf zum Beispiel A nicht Teil von B sein, wenn B Teil von A ist.

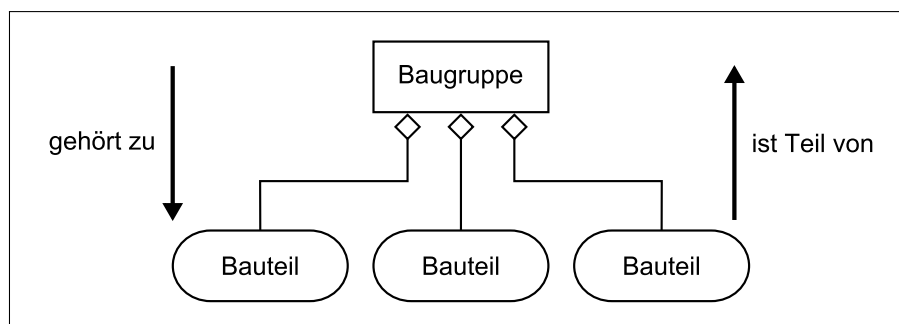


Abbildung 2.2: Aggregation in Bezug auf Produktstrukturen

Im Rahmen von Produktentwicklungsprozessen wird die Aggregation vor allem mit den Zusammenhängen zwischen Bauteilen und -gruppen eines Produktes in Verbindung gebracht. Aggregation bietet somit eine **Abstraktion** von Einzelbauteilen zu Bauteilgruppen und deren Eingliederung in Ober- und Untergruppen. Entgegengesetzt können

durch **Detaillierung**, konkrete Informationen gezielt abrufbar gemacht werden. Auf diese Weise ist es dem Anwender möglich, die Übersicht zu bewahren und schnelle Zugriffe auf Elemente zu erlangen.

2.2.2 Assoziation

Die **Assoziation** beschreibt, im Gegensatz zur Aggregation⁴, allgemeine Beziehungen bzw. Zusammenhänge zwischen zwei oder mehreren Elementen und Klassen. Im ERM werden hierbei gerichtete, ungerichtete, uni- und bidirektionale Assoziationen unterschieden. Typischerweise besitzen Assoziationen einen Namen und eine **Kardinalität**. Diese gibt an, welche Elemente einer Klasse wie und mit welcher Anzahl von Elementen einer in Verbindung stehenden Klasse assoziiert sind.

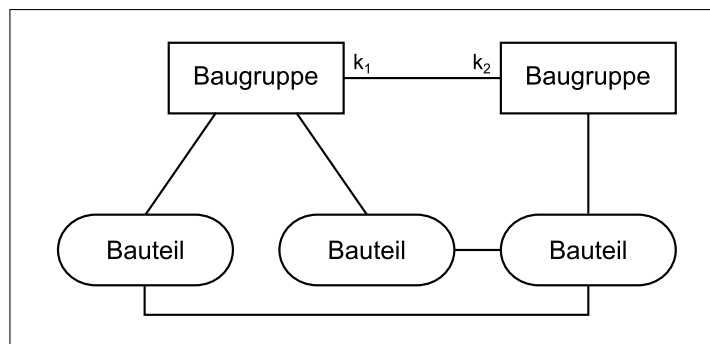


Abbildung 2.3: Assoziation in Bezug auf Produktstrukturen. Die Kardinalitäten können neben der Gerichtetheit auch Kann- oder Muss-Beziehungen wiedergeben. Beispielhaft dafür sind $k = 1$ genau eine Beziehung (muss), $k = *$ keine, eine oder mehr Beziehungen (kann), $k = 0..1$ keine oder eine Beziehung (kann), $k = 1..*$ eine oder mehrere Beziehungen (muss).

In der Produktentwicklung bilden Assoziationen simple Eigenschaftsverbindungen bis hin zu komplexen Netzwerkstrukturen. Diese dienen beispielsweise der Abbildung und Beschreibung von funktionalen Wirk- oder Bauzusammenhängen.

2.2.3 Variation

Variationen beschreiben verschiedene Ausführungsmöglichkeiten einer Struktur. Eine Strukturvariante ist somit eine konkrete Auswahl von optional zur Verfügung stehenden Element- und Klassenausprägungen, welche über UND-, ODER- und NICHT-Operatoren zu einer Gesamtstruktur komponiert werden. In der Produktentwicklung kann der Begriff

⁴ Die Aggregation stellt einen Sonderfall der Assoziation dar, bei dem nur hierarchische Beziehungen betrachtet werden.

daher wie folgt definiert werden [DIN02]:

„Varianten sind Gegenstände ähnlicher Form oder Funktion mit einem in der Regel hohen Anteil identischer Gruppen oder Teile.“

Primär gilt dies für **Bauteil- und Strukturvarianten**, welche sich entsprechend auf einzelne Bauteile oder Baugruppen beziehen. Unterschieden wird, ob die Variation der Bauteile Auswirkungen auf die Produktstruktur hat. Eine Strukturvariante würde daher zum Beispiel vorliegen, wenn statt einer Baugruppe ein einzelnes Bauteil verwendet wird oder durch eine Bauteilvariation die hierarchischen Zusammenhänge verändert werden.

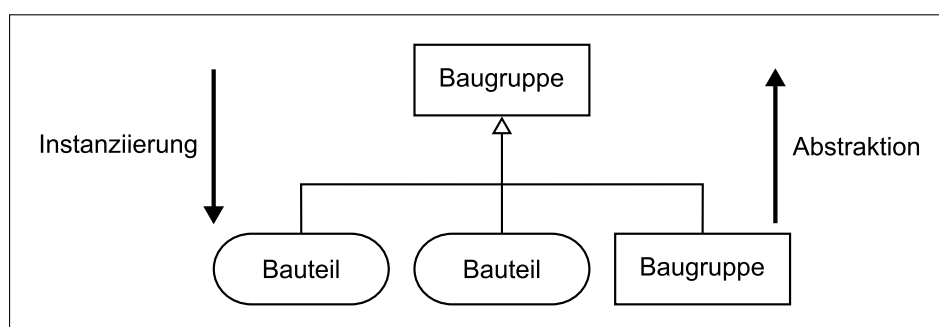


Abbildung 2.4: Variation in Bezug auf Produktstrukturen. Die Variation beschreibt zwingende und optionale Auswahl zwischen exklusiven oder inklusiven und disjunkten oder überschneidenden Alternativen.

Angesichts der heute vorherrschenden Marktgegebenheiten haben Produktvarianten eine hohe Bedeutung. Entwicklung und Zusammenstellung der Einzelteile und Baugruppen eines Produkts werden zunehmend den kundenspezifischen Anforderungen angepasst. Ein Beispiel dafür ist die Konzeption eines Fahrrads, welches je nach Kundenwunsch in verschiedenen Ausführungen, vom Citybike bis zum Mountainbike, individuell abstimmbare ist.

2.2.4 Veränderung

Veränderung beschreibt den Verlauf der Umwandlung bzw. Revision einer Struktur in Bezug auf die benötigte Zeiteinheit. Der Zustand einer Struktur zu einem bestimmten Zeitpunkt kann dabei anhand von **Änderungsständen** eindeutig definiert werden. Im Kontext des **Änderungsmanagements** werden die einzelnen Änderungsstände in einer Historie zusammengefasst, verwaltet und deren zeitliche Gültigkeit beschrieben. Die Änderungsstände können überdies **Versionen** zugeordnet werden, welche sich in Haupt- und Nebenversionen gliedern. Durch diese Form der Dokumentation werden Strukturentwicklung, zeitliche Gültigkeiten, aber auch spezifische Zustände mit deren Zusammensetzungen und Variationen gezielt abruf- und nachprüfbar.

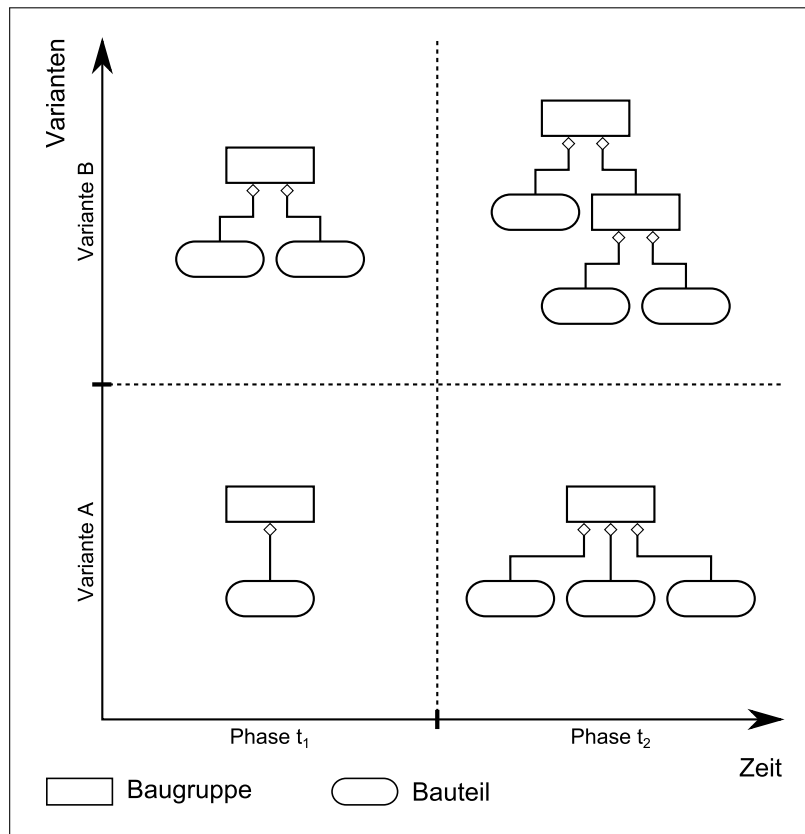


Abbildung 2.5: Änderungsstände und Varianten von Produktstrukturen

In der Produktentwicklung entstehen Veränderungen vor allem bei der Revision von Bauteilen und -gruppen. Diese haben zumeist gleichzeitig Änderungen an der Produktstruktur zu Folge. Zur Dokumentation der zeitlichen Gültigkeit und des Änderungsstandes werden daher spezielle Attribute wie Datumstempel oder Seriennummern eingesetzt.

2.3 Produktrepräsentation und -präsentation

Die Darstellung betreffender Produktinformationen eines Produktdatenmodells können im Wesentlichen in Geometriemodelle und Produktstrukturen unterschieden werden. Dabei beziehen sich die darzustellenden Informationen vorwiegend auf die bei der Produktentwicklung festgelegten Produktdefinitionen und anwendungsunabhängige rechnerinterne **Repräsentationsformen**⁵. Durch die **Produktpräsentation** können verschiedene Sichtweisen, wie Anforderungs-, Konstruktions-, und Dokumentationssicht, auf den Produktdatenbestand eingenommen werden. Je nach Informationsbedarf werden Produktdaten extrahiert und den Anwendern entsprechend bereitgestellt. Dadurch wird es für unterschiedlichen Gruppen möglich, die jeweiligen fachspezifischen Informationen zu betrachten und relevante Beziehungen und Modelle zu diskutieren.

⁵ Grundlegende Ansätze zur Geometrirepräsentation sind CSG, B-Rep und hybride Repräsentationsformen. Für Produktstrukturen existieren anwendungsneutrale Informationsmodelle, welche zusammen mit den topologischen Repräsentationsformen im Rahmen von STEP (ISO-Norm 10303) definiert sind.

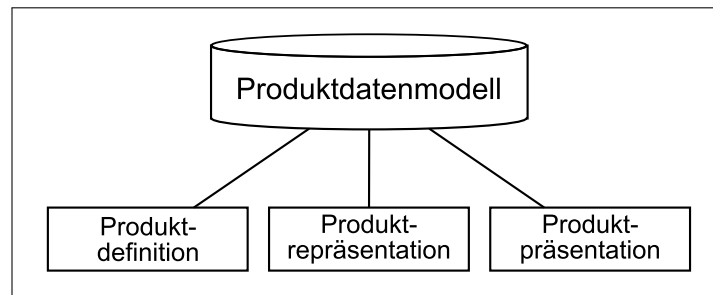


Abbildung 2.6: Repräsentation und Präsentation von Produktdaten

Die **Geometriepräsentation** bezeichnet die visuelle Abbildung von 3D-Geometriemodellen mittels Computergrafik (z.B. räumliche Darstellung, Animationen und virtuelle Realität) und von konventionellen technischen Zeichnungen und Dokumenten. Zu den herkömmlichen Präsentationsformen von **Produktstrukturen** gehören hingegen verschiedene Ausprägungen von Graphen und Bäumen, Stücklisten und Verwendungsnachweise. Hauptelemente von **Graphen** sind typischerweise Knoten und zwischen den Knoten liegende Kanten. Knoten entsprechen Bauteilen oder -gruppen eines Produktes und Kanten stellen die Beziehungen zwischen den Teilen dar. Die Kanten können dabei gerichtet (Aggregation, Assoziation) oder ungerichtet (Assoziation) sein und durch Querverweise netzartige Strukturen bilden. Im Gegensatz dazu sind **Bäume** spezielle Graphen, die sich auf ein hierarchisches Ordnungssystem beziehen. Dadurch sind hierarchische Beziehungen zwischen Bauteilen und -gruppen und zugeordnete Hierarchieebenen leichter ersichtlich. Beispiele für Baumpräsentationen sind Stammbäume, Aufbauübersichten und Erzeugnisgliederungen, welche die Gliederung des Gesamtproduktes in Bauteile, Haupt- und Unterbaugruppen beschreiben. So wird die Aufbauübersicht nach [DIN03] wie folgt definiert:

„Aufbauübersicht ist eine bevorzugt grafisch angeordnete Übersicht über die Angliederung eines zusammengesetzten Gegenstandes (Produkt oder Unterbaugruppe).“

Zu den **Stücklistenarten** gehören zum Beispiel Konstruktions- und Fertigungsstücklisten und Varianten- oder Baukastenstücklisten, welche als Hauptmerkmal einen tabellarischen Aufbau gemein haben. Der Verwendungszweck von Stücklisten liegt in der Definition und Dokumentation des Produktaufbaus und der Identifizierung von Mengen und Einheiten begründet. In diesem Kontext dienen **Verwendungsnachweise** ferner dazu, Transparenz für die Verwendung von Produkten und Produktteilen zu schaffen.

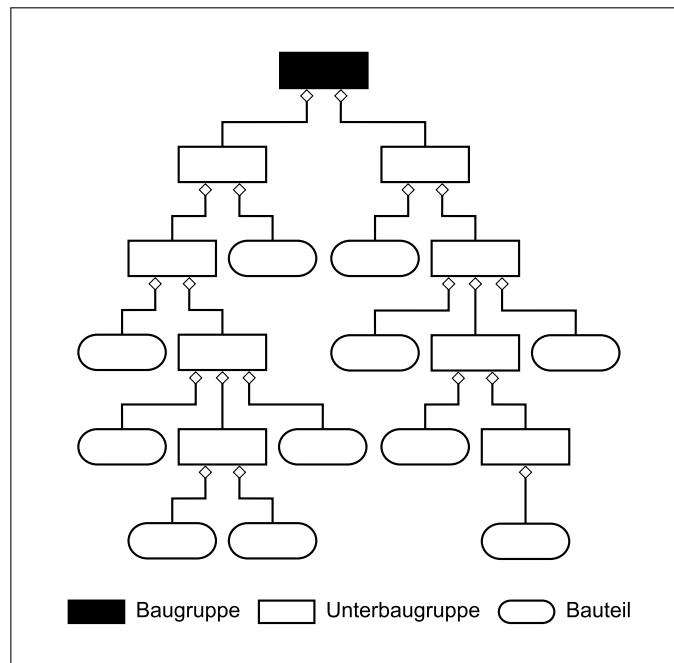


Abbildung 2.7: Bäume als Produktstrukturpräsentation. Abgebildet ist ein Produktstammbaum mit Baugruppen und Bauteilen.

2.4 Visualisierung und Data Mining

2.4.1 Visualisierung

Die **Visualisierung** beschreibt im Allgemeinen die Abbildung von abstrakten Daten und Zusammenhängen in eine grafische bzw. visuell erfassbare Form und wird wie folgt definiert [VDI03]:

„Die Visualisierung umfasst die Erzeugung der grafischen Veranschaulichung von Daten und Sachverhalten durch Transformation ermittelter Rohdaten in symbolische und geometrische Information.“

Abstraktion der Ausgangsdaten, geeignete Gestaltungsentscheidungen und implizierte Interpretation sind bei der Visualisierungskonzeption ausschlaggebend. Die Visualisierung teilt sich in verschiedene Bereiche und erfordert zumeist eine fachübergreifende Zusammenarbeit von verschiedenen Forschungsfeldern. Im wissenschaftlichen Sinne werden so Kenntnisse aus Farbenlehre, Biologie (Aufbau des menschlichen Auges), Psychophysik und kognitiver Psychologie genutzt. Des Weiteren haben je nach Anwendungsgebiet auch Ingenieurs- und Informatikwissenschaften einen hohen Stellenwert. Die daraus entstehenden Disziplinen sind unter anderem das Cognitive Engineering, Visual Engineering und Softwareergonomie. Arbeiten in diesem Bereich beziehen sich dabei nicht nur auf die Aufbereitung und grafische Darstellung von Daten, sondern auch auf die Anpassung von Anwendungssystemen an menschliche Fähigkeiten und Wahr-

nehmung. Anwendung und Ziele der Visualisierung sind, den Wahrnehmungsaufwand bei der Analyse großer Datenbestände zu verringern und Informationen ansprechend darzustellen und realisierbar zu machen. Demnach können Funktion und Eigenschaft auch folgendermaßen definiert werden [Tuf01, Tufte S. 51]⁶:

„Grafische Exzellenz verleiht dem Betrachter größtmögliches Verständnis, in der kürzesten Zeit, mit geringster Tinte und auf kleinstem Raum.“

Grundsätzlicher Anspruch ist, neben einer hohen Anschaulichkeit des Visualisierungsmodells, auch die Bildung einer gemeinsamen Kommunikationsgrundlage für die beteiligten Benutzer. Die Visualisierung hat somit einen hohen Stellenwert als interdisziplinäres Kommunikationsinstrument und findet im Unternehmensumfeld, wie in Tabelle 2.2 am Beispiel der Simulation dargestellt, in verschiedensten Funktionseinheiten und Einsatzfeldern Verwendung. In vielen Bereichen ist die Visualisierung die Schnittstelle vom Datenmodell zum Benutzer, wobei sich die Art der Darstellung von Informationen nach den Bedürfnissen der Zielgruppe richtet.

Einsatzfelder	Zielgruppen							
	Fachabteilung	Planung	Management	Technischer Einkauf	Werktechnik und -vertrieb	Produktion und Logistik	Vertrieb und Marketing	Kunde Öffentlichkeit
Systemanalyse	●	●		●				
Modellerstellung	●	○						
Validierung	●	●						○
Experimentdurchführung	●	●						○
Ergebnisdarstellung und Interpretation	●	●	●	●	●	●	●	●
Schulung						●	○	●
Betriebsbegleitung		●	○			●		○
Absatzförderung und Außendarstellung			●				●	● ●
Interdisziplinäre Kommunikation	●	●	●	●	●	●	●	●

● voll zutreffend ○ teils zutreffend

Tabelle 2.2: Relevanz der Visualisierung in Bezug auf Zielgruppen und Einsatzfelder im Umfeld der Simulation nach [VDI03]

Das Prinzip der Visualisierung wird vor allem durch das menschliche Sehvermögen begünstigt. So gehören unter anderem Mustererkennung, Übertragungsbandbreite der

⁶ „Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space.“

oder Ergebnisse. Die resultierende Interpretation ist demnach von diesen und weiteren Kriterien abhängig, wodurch unter Umständen auch ungewollte oder gar vorsätzliche Trugbilder⁸ entstehen können. Wichtige Prinzipien (nach Tufte⁹) für den Entwurf von Visualisierungen sind demnach:

- Darstellung der Daten
- Vermeidung von Verzerrung und Verfälschung der Datenaussage
- Präsentation vieler Werte auf kleinem Raum
- Folgerichtigkeit großer Datensätze
- Förderung von Vergleichen unterschiedlicher Datenteile
- Darlegung verschiedener Detaillierungsgrade

Die statische Visualisierung ist in ihrem Potenzial begrenzt und auf die Veranschaulichung von im Vorfeld extrahierten und kompilierten Informationen limitiert. Das Feld der **Informationsvisualisierung** hingegen impliziert die Interaktion mit der Darstellung durch den Benutzer und wird definiert [SJ07, Card S. 542]¹⁰:

„Informationsvisualisierung ist der Einsatz von Technologien, welche mittels Computervisualistik die menschliche Wahrnehmung anhand abstrakter Informationen erweitern.“

Unter abstrakten Informationen werden dabei jegliche grafische Elemente und Objekte verstanden, die zwangsläufig keine direkte physische Relationen besitzen. Wahrnehmungserweiterung ist in diesem Zusammenhang Effizienzsteigerung und Kenntnis über relevante Muster und flexibler Umgang mit den kodierten Informationen. Somit können oben angeführte Prinzipien ebenso auf die Konzeption bei der Informationsvisualisierung angewandt werden. Besondere Herausforderungen sind dabei hohe Dimensionalität der Daten, Skalierbarkeit, erweiterte Filter- und Bedientechniken und die Evaluation der Gebrauchstauglichkeit.

Die innerhalb der Visualisierung verwendete **Repräsentation** definiert die Form der visuellen Beschreibung von Informationen. Hierbei können folgende Ausprägungen unterschieden werden:

⁸ E. Tufte definierte hierfür 1983 den Lie Factor (Lügenfaktor), welcher den Zusammenhang zwischen Ausmaß des Effekts in einer Darstellung und des tatsächlichen Effekts in den zugrundeliegenden Daten beschreibt.

$$\left[\text{Lie Factor} = \frac{\text{Effekt in Darstellung}}{\text{Effekt in Daten}} \right], \text{ wobei } \left[\text{Effekt} = \frac{|\text{Zweiter Wert} - \text{Erster Wert}|}{\text{Erster Wert}} \right]$$

⁹ Edward R. Tufte war bis zu seiner Emeritierung 2004 Professor für Statistik, Grafikdesign und politische Wissenschaft in Yale und gilt als Fachmann für die Gestaltung und Präsentation von Informationsgrafiken.

¹⁰ „Information visualization is a set of technologies that use visual computing to amplify human cognition with abstract information.“

symbolisch bezieht sich auf die Verwendung von Symbolen deren Bedeutung mittels eines Codes festgelegt ist. Es werden Zeichen und allgemeine abstrakte Symbole unterschieden. Zeichen sind Elemente zur Darstellung von Informationen aus einer vereinbarten endlichen Menge, wie alphanumerische oder numerische Zeichen. Abstrakte Symbole wiederum haben eine eingeschränkte, zum Beispiel auf den Anwendungsbereich festgelegte, Bedeutung und stellen einfache 2D-, 2½D- oder auch 3D-Symbole dar. Zudem können abstrakte Symbole auch als Piktogramme bezeichnet werden, wenn die Voraussetzung einer international festgelegten Bedeutung (zum Beispiel Totenkopf für Gift) erfüllt ist.

ikonisch beschreibt den Einsatz von Ikons (Bildzeichen), welche unmittelbar die eigene Bedeutung repräsentieren und im Allgemeinen keine Code-Konvention voraussetzen. Durch die Ähnlichkeit zum anschaulichen Original, wird dem Betrachter demnach sofort deutlich, was vermittelt wird. Es werden stilisierte und realitätsnahe Formen unterschieden, welche anhand der visuellen Merkmale einen gewissen mehr oder weniger starken Realitätsbezug darstellen. In Analogie zu abstrakten Symbolen können Ikons durch Konventionalisierung zu Piktogrammen mit international festgelegter Bedeutung werden (zum Beispiel stilisierte Abbildungen von Sportarten).

fotorealistisch sind Repräsentationsformen (realistische Abbildungen), die im höchsten Maße Anforderungen an die computer-generierbare Realitätsnähe stellen. Dabei wird ein Ausschnitt der Realität unter Berücksichtigung physikalischer Gegebenheiten, wie Reflexion und Spiegelung, nahezu 1:1 wiedergegeben.

Im Umfeld der Informationsvisualisierung werden kodierte Informationen auch als **Glyphen** (Bildzeichen) bezeichnet. Dies sind grafische Einheiten, die multiple Datenwerte anhand leicht unterscheidbarer Attribute und deren Zusammensetzung und Kombination repräsentieren. Die Erscheinung der Objekte wird dabei mittels Richtlinien, wie bis zu 8 Farben, 4 Ausrichtungen, 4 Skalierungen und binäres Blinken, Texturen und Formen bestimmt.

Die **Darstellungsform** eines Visualisierungsverfahrens beschreibt die Gesamtwiedergabe des Sachverhaltes unter Nutzung der genannten Repräsentationsformen. Dies umfasst den Einbezug und die Zusammenstellung verschiedenster Elemente und Formen, wie Schrift, Tabellen, Diagramme, Zeichnungen und Bilder, die sich in ihrer Art, Aussagekraft und Eignung für die Abbildung von unterschiedlichen Daten unterscheiden. Ferner kommen dabei Darstellungsformen der virtuellen und erweiterten Realität zum Einsatz.

Der Anwender als Betrachter oder Analytiker, dessen Handlungen und zur Verfügung gestellte Werkzeuge und Funktionen, spielen im Kontext der Visualisierung eine ebenfalls wichtige Rolle. Deren Anforderungen werden im Mantra der visuellen Informationssuche „*zuerst Übersicht, Zoom und Filter, dann Details auf Abruf*“¹¹ [Shn96] konkretisiert und

¹¹ Visual Information-Seeking Mantra „*Overview first, zoom and filter, then details on demand*“.

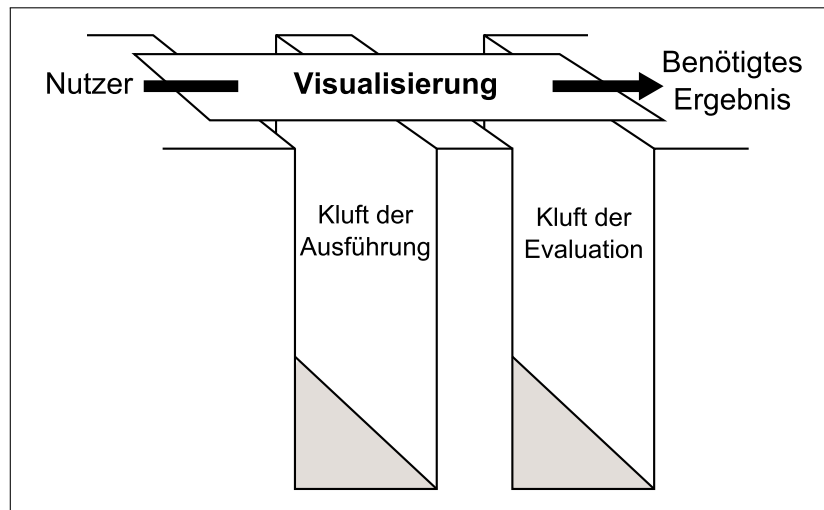


Abbildung 2.9: Kluft der Ausführung und Evaluation nach [PHP03]. Die Kluft der Ausführung bezeichnet wie eine Frage angegeben werden kann. Die Kluft der Evaluation bezieht sich auf die Interpretation des Ergebnisses.

stellen ein nützliches Bezugssystem für Funktionen, Aufgaben und Richtlinien für die Entwicklung einer Anwendung für die Informationsvisualisierung dar. Eine genauere Beschreibung der Funktionen, zu denen in [Shn96] auch die Anzeige von Zusammenhängen, Aktionshistorie und die Extraktion und Klassifizierung von Teilmengen zählen, wird in Kapitel 2.5 eingehend diskutiert.

Bei der Arbeit mit einer Visualisierungsanwendung ist der Benutzer dennoch oftmals mit einem nur bedingt intuitiven System konfrontiert. Der Informationsgehalt einer Darstellung kann zwar mit der Aussage „*ein Bild sagt mehr als Tausend Worte*“ beschrieben werden, jedoch wurde hierzu im Bereich der Informationsvisualisierung von [PHP03, Larkin & Simon, S. 1] qualitativ angemerkt „*eine Grafik sagt manchmal mehr als Tausend Worte*“. Diese Aussage bezieht sich auf die Art und den Kontext der Informationspräsentation. Die Problematik besteht folglich zumeist im Wissen wie eine Anwendung veranlasst werden kann, konkrete Aufgaben zu lösen und wie anschließend die resultierende Präsentation zu interpretieren ist. Dies liegt darin begründet, dass generell davon ausgegangen werden kann, dass der Benutzer keine Vorstellung hat, welche Mittel ihm zur Verfügung stehen und wie diese einsetzbar sind. Dieser Umstand ist in Abbildung 2.9 dargestellt. Im Prozess der Ergebniserzeugung durchläuft der Benutzer daher die Phasen der Formulierung, Aktionsinitialisierung, Ergebnisrevision und Verfeinerung. Diese Schritte werden bei der Analysearbeit meist iterativ durchlaufen, bis genügend wesentliche Informationen ausgewertet oder Endresultate erreicht sind. Das System muss dabei in engem Dialog mit dem Benutzer stehen und ihn bei dessen Handlungen sinnvoll unterstützen, so dass die kognitive Belastung auf relevante Informationen und Aktionen beschränkt bleibt. Eine umfassende Übersicht und beeinflussende Faktoren einer Systematik für Entwicklung und Konzeption der Informationsvisualisierung wird abschließend in Abbildung 2.10 veranschaulicht.

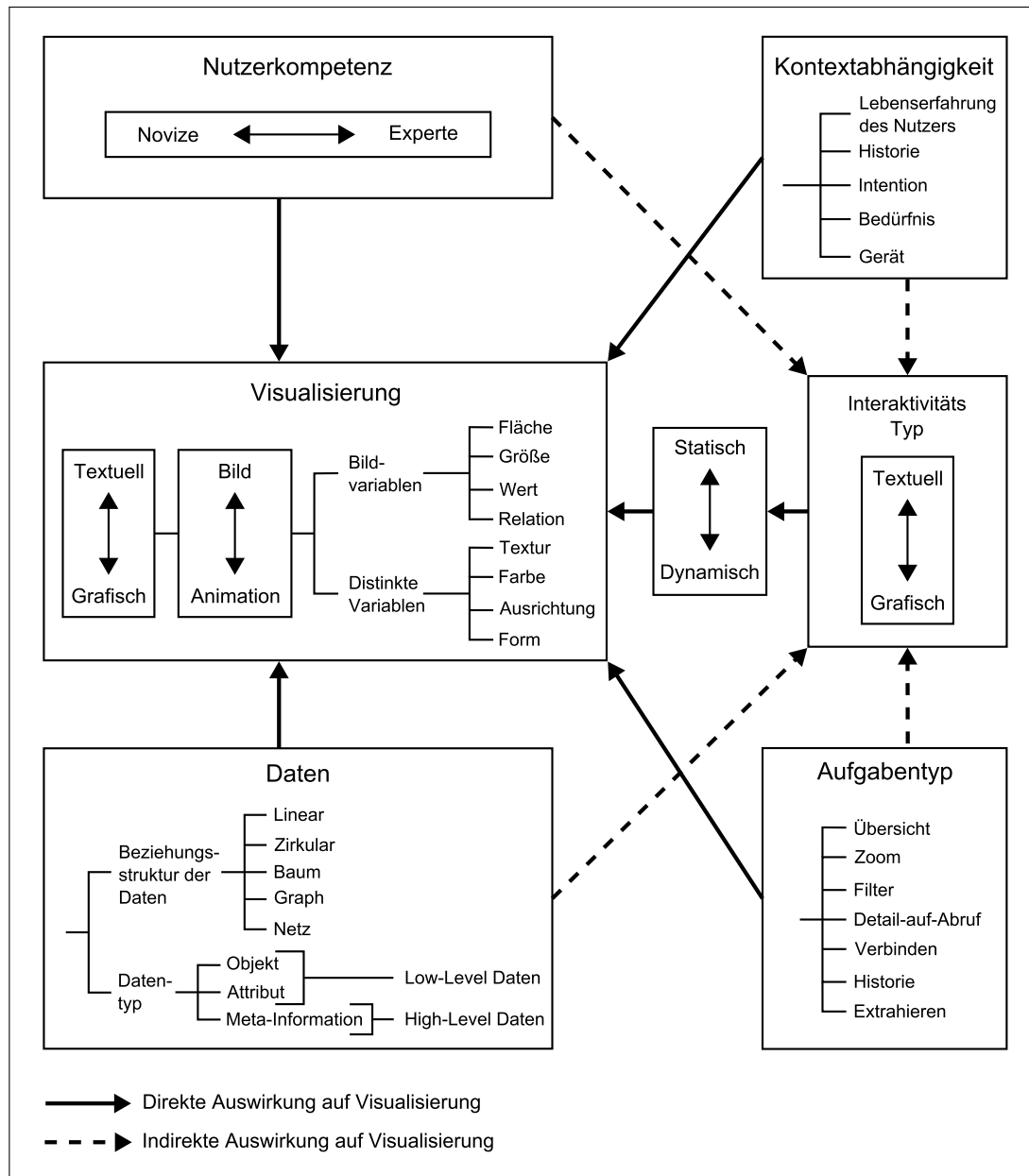


Abbildung 2.10: Ein Bezugssystem für die Taxonomie der Informationsvisualisierung nach [PHP03]

2.4.2 Visualisierungspipeline

Beim Visualisierungsprozess gilt es abstrakte Daten, meist ohne geometrische Natur, in Form von Bildern zu veranschaulichen. Die bei diesem Prozess zu durchlaufenden Schritte werden in die **Visualisierungspipeline** eingeordnet [SM00]. Die Visualisierungspipeline definiert eine Abfolge von geschalteten Funktionen, die zum Generieren, Filtern und Bereinigen, Abbilden auf Geometrien, Rendern und Darstellen von Informationen dienen. Als Erweiterung wird meist die interaktive Ausführung der Steuerung durch den Anwender mit eingeschlossen. Die Prozesskette der Visualisierungspipeline kann in die drei grundlegenden Phasen Datenaufbereitung, Datenmapping und Bildgenerierung unterteilt werden.

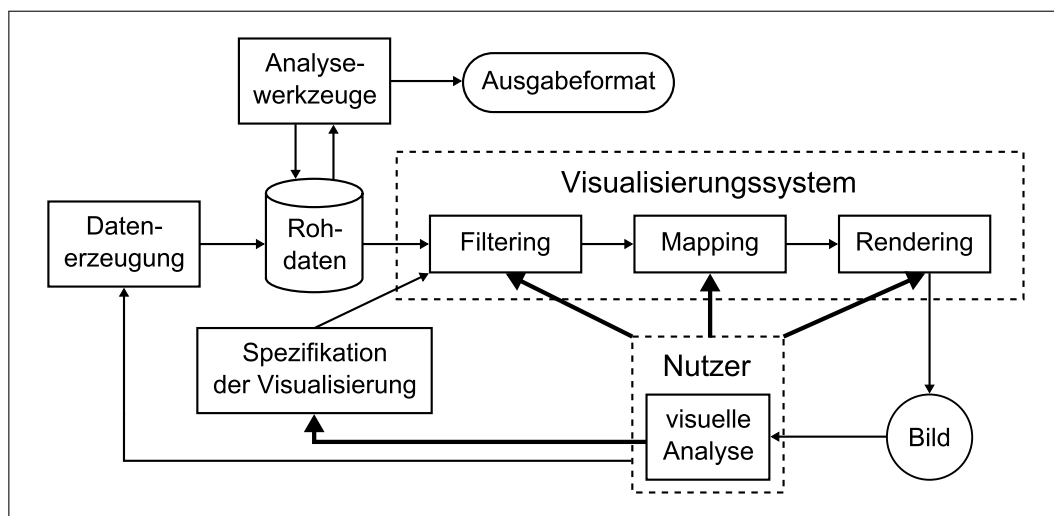


Abbildung 2.11: Ein integriertes Referenzmodell für die Visualisierung nach [SM00]. Der Benutzer ist hier im Visualisierungsprozess direkt involviert und übernimmt gleichzeitig die Rolle des Betrachters und des Visualisierungsautors.

Die **Datenaufbereitung** realisiert eine Rohdaten-zu-Nutzdaten-Abbildung. Rohdaten bezeichnen dabei die von einem IT-System erhobenen Daten, welche für die nächsten Visualisierungsschritte aufbereitet werden. Grundoperationen sind dabei Vervollständigung oder Reduzierung und eine etwaige Konvertierung von relevanten Daten. Ein weiterer wichtiger Bestandteil ist das Filtern der Daten, wobei Extraktionen, Fehlerkorrektur und Glättung von Datenwerten vorgenommen werden. Somit sind neben einer direkten 1:1-Abbildung, d.h. der vollständigen Überführung der Ausgangsdaten, weitere Formen der Datenübernahme und deren Kombinationen möglich. Visuelle Abstraktion beschreibt in diesem Zusammenhang die Fokussierung und Reduktion der Datenmenge durch Projektion, Verringerung der darzustellenden Variablen durch Strukturänderung und Selektion durch Filterung. Die visuelle Detaillierung hingegen bezieht sich auf Inter- oder Extrapolation der Datenmenge durch Ergänzung zusätzlich visueller Details. Visuelle Erweiterung ist letztlich die Hinzunahme von nicht zwangsläufig zu untersuchenden Sachverhalten und Elementen, welche zum Beispiel den Kontext der Daten wiederge-

ben. Das Ergebnis bilden aufbereitete Daten, die anschließend für folgende Schritte zur Verfügung stehen.

Die Erzeugung eines Grafikmodells wird durch das **Datenmapping** beschrieben. Es wird in hohem Maße Einfluss auf die visuelle Darstellung der Daten genommen und Layout- und Integrationsfragen zu bereits vorhanden grafischen Darstellungen diskutiert. Fundamental ist des Weiteren die Auswahl der Darstellungselemente und deren Belegung mit Datenwerten, welche vor allem Auswirkungen auf Expressivität und Effektivität der entstehenden Darstellung haben.

Die letzte Phase ist die **Bildgenerierung** (Rendering), wobei eine Abbildung der Darstellungselemente auf Bilddaten stattfindet. Durch Bildverarbeitungssysteme werden aus der Grafikbeschreibung statische oder auch dynamische Bilder erzeugt. Je nach Anwendungsumfeld unterscheidet sich dabei die Eignung dieser Darstellungsarten, welche meist im Datenmapping vordefiniert werden. Mögliche Beispiele für Darstellungsarten sind:

Realitätsnahe Bilder versuchen den Beobachtungsraum, aus dem die Daten gewonnen wurden, möglichst realistisch in die Darstellung einzubeziehen oder schaffen eine Abbildung der Daten in einer natürlichen Szene.

Abstrahierende Bilder generalisieren Details und veranschaulichen Daten durch expressive Darstellungselemente.

Mentale Bilder kombinieren äußere und innere Strukturen und lassen damit oft auch verborgene Zusammenhänge sichtbar werden.

Animationen sind sich kontinuierlich über die Zeit verändernde Bilder.

Der gesamte Prozess kann von einem **Visualisierungsautor** betreut werden, welcher den Datenfluss steuert und die Visualisierung betreffende Entscheidungen trifft. Der Autor kann die Schritte der Visualisierungspipeline (Filtering, Datenmapping, Rendering) direkt kontrollieren und somit Einfluss auf die entstehende Darstellung ausüben. Ferner kann der **Betrachter**, je nach Vertrautheit mit dem System, ebenfalls selbständig oder in Zusammenarbeit mit dem Autor auf den Prozess einwirken. Dadurch ist es möglich, dem Betrachter genau spezifizierte Darstellungen nach konkreten Präferenzen und Bearbeitungszielen für die visuelle Analyse zur Verfügung zu stellen.

Bei der Konzeption der Visualisierungspipeline wird oftmals auch implizit die Implementierung als Visualisierungsprogramm auf einem Computer besprochen. Durch dessen Verwendung wird das automatisierte Finden und Bewerten von Datenmustern als Bestandteil des Data-Mining komplettiert. Die Methodik der Visualisierungspipeline ist allerdings allgemein formuliert und bedarf in der Regel einer Anpassung an den jeweiligen Anwendungskontext. Für Produktstrukturen sind hier vor allem die Besonderheiten der elementaren Ausprägungen und deren Kompositionsmöglichkeiten mit den Geometrie-modellen zu berücksichtigen.

2.4.3 Data Mining

Generell beschreibt der Begriff **Data Mining**¹² die Analyse von großen Datensätzen unter der Betrachtung von verschiedenen Perspektiven und die Zusammenfassung und Extraktion relevanter Informationen. **Daten** sind jegliche Art von digitalen Formaten, wie Fakten, Nummern, Texte oder Bilder, die mit IT-Systemen bearbeitet werden können. Es wird zwischen operationalen, transaktionalen Daten (permanent bearbeiteten, aktuellen Daten), nonoperationalen Daten (nicht aktuellen, archivierten Daten) und Metadaten unterschieden. **Informationen** beschreiben hingegen für den Betrachter relevante Muster, Assoziationen und Zusammenhänge, welche aus der gesamten Datenmengen extrahiert werden. Als letzten Schritt im Prozess wird die Information idealerweise in **Erkenntnis** bzw. Wissen umgewandelt, wodurch Rückschlüsse auf historische Muster oder zukünftige Entwicklungen getroffen werden können.

Häufige **Anwendung** findet Data Mining bei der Untersuchung großer elektronischer Datenbestände, deren Anzahl fortwährend rapide ansteigt. In diesem Zusammenhang wird mit dem Begriff der Datenexplosion die Beobachtung bezeichnet, dass die Menge an Daten in der Informationsgesellschaft gegenüber anderen Bereichen überproportional anwächst. Im Jahre 2003 wurde von der Universität in Berkeley zu diesem Thema eine Studie durchgeführt [LVC03]. Als Ergebnis wurde unter anderem festgestellt, dass jeder Mensch beinahe 800 Megabyte (MB) an digitalen Daten pro Jahr produziert, und dass allein im Jahr 2002 ca. 5 Exabyte (EB)¹³ an Daten (Film, Druck, magnetische und optische Speichermedien) produziert wurden. Zu beachten ist, dass dabei hauptsächlich die Quantität der Daten, nicht aber zwangsläufig die Qualität gemeint sind. Die Problematik besteht daher sowohl für Einzelne, als auch für Unternehmen weniger in der Verfügbarkeit von Daten, sondern vielmehr in der Beherrschung der Datenflut. Dies führt zu steigendem Interesse an Data Mining und der Notwendigkeit von Methoden, um die Daten automatisiert zu sortieren und wesentliche Informationen zu filtern.

Der Prozess der Erkenntnisgewinnung aus Datenbeständen besteht aus einer iterativen Sequenz der folgenden elementaren Schritte [HK06]:

1. **Datensäuberung** entfernt beeinträchtigende und verzerrende Elemente sowie inkonsistente Daten.
2. **Datenintegration** kombiniert gegebenenfalls multiple Datenquellen.
3. **Datenselektion** ruft analyserelevante Daten aus der Datenbank ab.
4. **Datentransformation** vereinigt und konvertiert Daten in eine Data Mining geeignete Form, zum Beispiel durch Summierung oder Aggregation.
5. **Data Mining** umfasst den essentiellen Prozess der Anwendung von intelligenten Methoden, um Datenmuster zu extrahieren.

¹² Auch Knowledge Discovery in Databases (KDD) bzw. Erkenntnisgewinnung aus Datenbeständen.

¹³ Ein Exabyte entspricht einer Million Terabyte, was in etwa äquivalent der Daten von 17 Millionen Bücher ist.

6. **Muster Evaluation** identifiziert Muster von tatsächlicher Relevanz anhand von wissens- und interessenbasierten Messkriterien.
7. **Erkenntnispräsentation** ist die Verwendung von Visualisierungs- und Wissenspräsentationstechniken, um dem Benutzer die gewonnenen Kenntnisse zu vermitteln.

Die Schritte 1 bis 4 sind unterschiedliche Formen der Datenvorverarbeitung, um die Daten für den Data Mining Prozess zu präparieren. In der Informationsindustrie werden dabei Datensäuberung und -integration häufig kombiniert und die Resultate in einem Data Warehouse¹⁴ für weitere Schritte abrufbar bereitgehalten. Außerdem wird hierzu die Datentransformation fallweise vor der Datenselektion ausgeführt und optionale Datenreduktionen, ohne die Datenintegrität zu vermindern, vorgenommen. Generell interagiert der Data Mining Prozess direkt mit dem Benutzer oder mit einer Wissensbasis, wobei interessante Informationen dem Benutzer präsentiert und als neue Bestandteile in der Wissensbasis abgelegt werden können.

In Abbildung 2.12 ist ein typisches Data Mining System, unter Berücksichtigung des beschriebenen Prozesses zur Erkenntnisgewinnung, dargestellt. Dabei können die folgenden Hauptkomponenten unterschieden werden [HK06]:

Datenbanken, Data Warehouse und andere Datendepots sind Bestände von Datenbanken, Data Warehouses, Tabellen und Arbeitsblättern. Diese und weitere Arten von Datenablagen bilden die Basis für die Datensäuberung und -integration.

Datenbank- oder Warehouse Server sind für den Abruf relevanter Daten anhand von Nutzeranfragen verantwortlich.

Wissensbasis ist das Fachwissen, welches die Informationssuche leitet oder die interessenbasierte Evaluation von resultierenden Mustern unterstützt. Derartiges Wissen beinhaltet zum Beispiel Hierarchiekonzepte, welche Attribute und Datenwerte in verschiedene Abstraktionslevel organisieren. Dabei ist der Einbezug der Nutzerüberzeugung für die Bemessung der Relevanz bzw. Informativität¹⁵ von Mustern möglich. Weitere Beispiele für Nutzung von Fachwissen ist der Einsatz ergänzender Bedingungen, Grenzwerte und Metadaten.

Data Mining Maschine ist die essenzielle Hauptkomponente für das Data Mining System und besteht idealerweise aus funktionalen Modulen für Charakterisierung, Assoziation und Korrelationsanalyse, Klassifizierung, Prognose und Cluster-, Ausreißer- und Evolutionsanalyse.

Muster-Evaluation-Modul ist für die Messung der Informativität zuständig und interagiert mit der Data Mining Maschine, um den Fokus der Informationssuche auf relevante Muster zu lenken. Bereits erkannte Schemata werden dabei anhand von Grenzwerten gefiltert. Alternativ kann das Evaluationsmodul auch mit der Data

¹⁴ Ein Data Warehouse bzw. Datenlager ist eine zentrale Datensammlung, zum Beispiel eine Datenbank.

¹⁵ Informativität ist das Ausmaß der Erwartetheit bzw. Unerwartetheit von dargebotenen Mustern. Muster sind genau dann informativ, wenn diese für den Rezipienten neue Informationen enthalten.

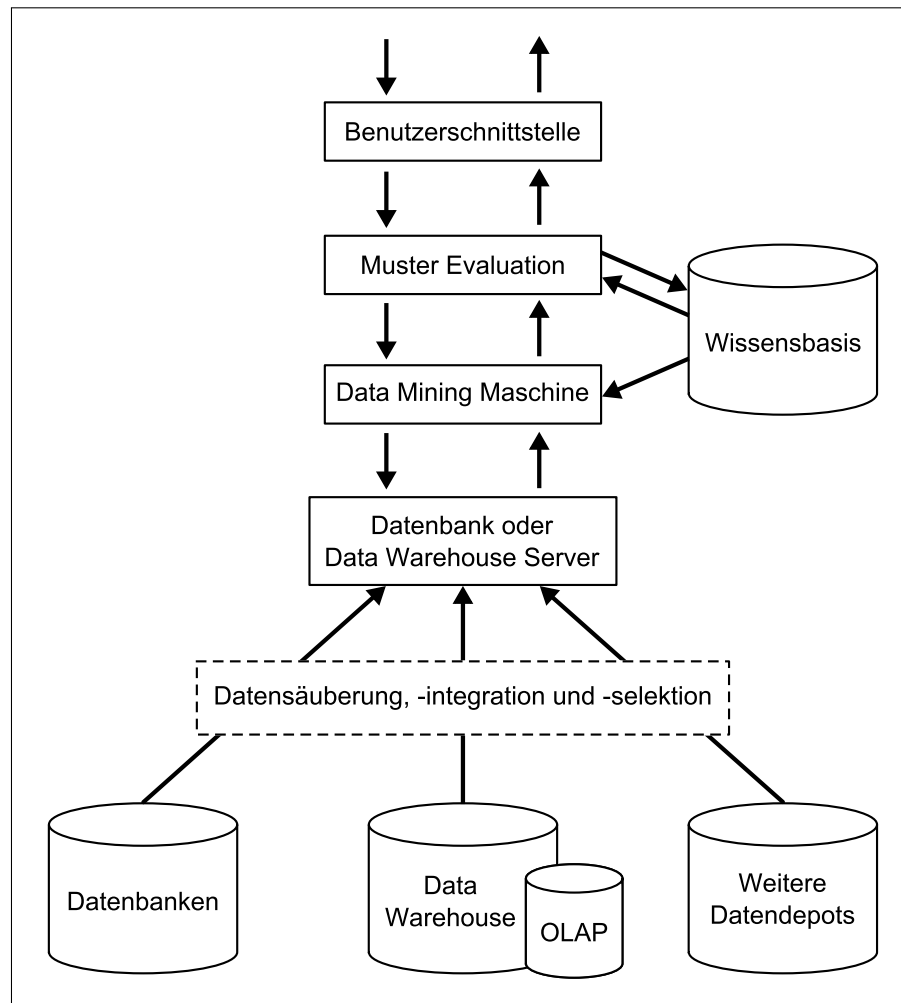


Abbildung 2.12: Architektur eines typischen Data Mining Systems

Mining Komponente verbunden werden, um die Gesamteffizienz zu steigern und die Informationssuche auf relevante Daten abzugrenzen.

Benutzungsschnittstelle erlaubt dem Benutzer die Kommunikation und Interaktion mit dem Data Mining System. Der Benutzer kann spezielle Anfragen stellen und wird mittels zusätzlicher Informationen und Zwischenergebnisse bei der Suche und dem Erkunden des Datenbestandes unterstützt. Des Weiteren kann der Benutzer Data Warehouse-, Datenbank-, Datenstrukturschemata überprüfen, erkannte Muster auswerten und in verschiedenen Formen visualisieren.

Der schon angeführte Begriff des **Data Warehouse** bezeichnet ein unverteiltes Depot von Daten. Die Daten können aus verschiedenen Quellen stammen und werden zentral nach einem einheitlichen Schema abgespeichert. Entstehung und Erhaltung eines Data Warehouses umfasst vornehmlich Datenaufbereitung, -integration, -transformation, -ladung und Aktualisierung, wobei abzulegende Daten typischerweise Hauptthematiken zugeordnet werden. Wie in Abbildung 2.12 und 2.13 dargestellt, fungiert das Data Warehouse somit als Verbindung zwischen erzeugenden Datenbanken und dem Data Mining Prozess. Konkret stellen Data Warehouses dabei meist Sekundärdatenbanken dar, in

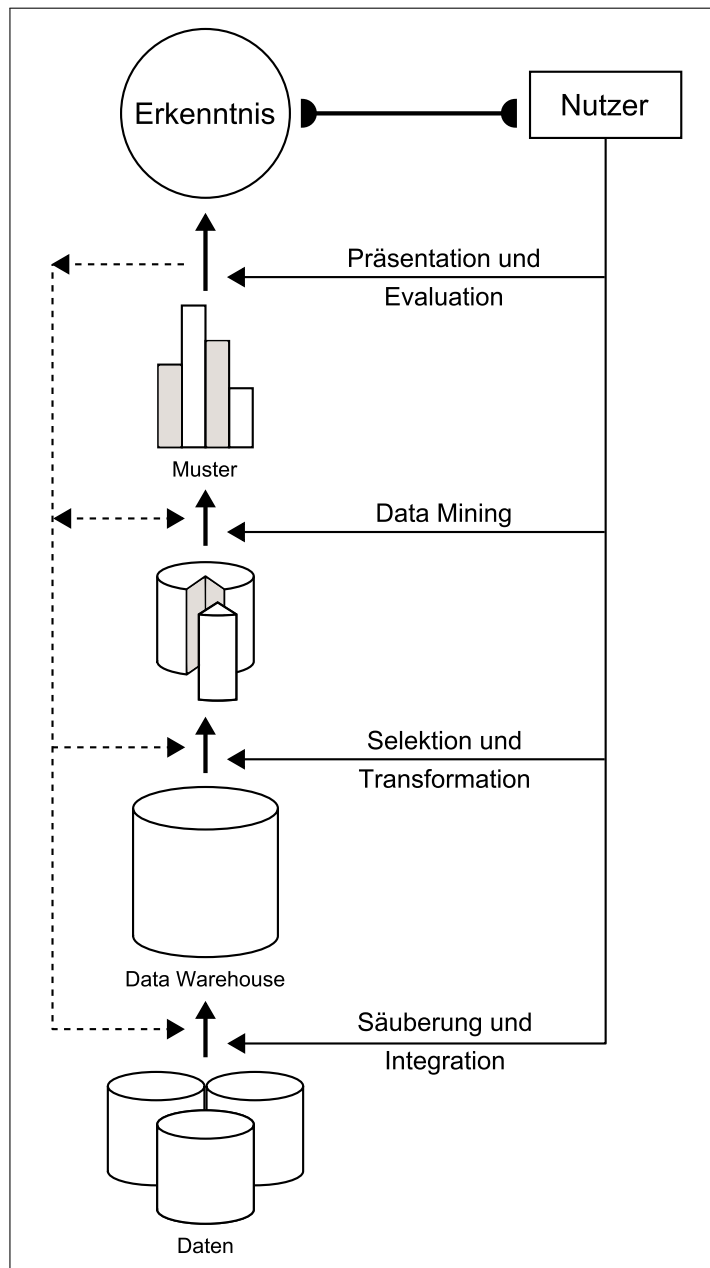


Abbildung 2.13: Data Mining als Bestandteil im Prozess der Erkenntnisgewinnung aus Datenbeständen nach [HK06]

denen Daten aus verschiedenen Primärdatenbanken abgebildet werden. Zur Verwaltung relationaler Datenbanken werden dafür Datenbankmanagementsysteme (DBMS) eingesetzt oder für spezielle Anforderungen auch multidimensionale Datenwürfelsysteme, welche Vorberechnungen und schnelle Zugriffszeiten erlauben. Durch den systematischen Aufbau sind Data Warehouses insbesondere für den Einsatz mit Online Analytical Processing (OLAP) geeignet. OLAP Funktionen verwenden Hintergrundwissen und Metadaten über die zu untersuchenden Daten, um diese in verschiedenen Abstraktionsstufen darstellen zu können. Je nach Nutzersicht können somit zum Beispiel unterschiedliche Detaillierungsgrade zeitlicher Zusammenfassungen von bestimmten Daten abgefragt werden. Nachteilig ist dabei, dass mit der Durchführung komplexer Analysevorhaben unter anderem hohe Datenaufkommen und Performanceprobleme verursacht werden.

Wie in Abbildung 2.13 ersichtlich, lässt sich Data Mining konkret in den Prozess der Erkenntnisgewinnung einordnen. Die einzelnen Schritte erinnern dabei stark dem in Kapitel 2.4.2 vorgestellten Visualisierungsprozess. Tatsächlich sind Data Mining und Visualisierung meist eng miteinander verbunden und beinhalten zusammenhängende und aufeinander aufbauende Abläufe. Die Visualisierung kann daher in Form der Evaluations- und Präsentationsphase auch als Teil der Erkenntnisgewinnung und Ergebnis einer Data Mining Analyse betrachtet werden.

Bei dem Umgang mit Produktstrukturen werden eine Vielzahl unterschiedlicher IT-Systeme und zugehörige Datenbanken eingesetzt. Das Management der entstehenden Datenmengen ist eines der Hauptfelder in heutigen Produktentwicklungsprozessen. Um zum Beispiel eine effiziente Visualisierung der elementaren Ausprägungen einer Produktstruktur zu realisieren, ist der Prozess des Data Mining (Datentransformation, Datenaufbereitung, Präsentation, etc.) essentiell. Zusätzlich kann der Einsatz von Data Warehouses in diesem Zusammenhang durch die systematische Vorgehensweise der Generierung und Haltung von relevanten Daten zu einer erheblichen Komplexitätsreduktion führen.

2.5 Interaktion und Navigation

Die Visualisierung bezieht sich im informationstechnischen Bereich auf die bildliche Darstellung von Datenrepräsentationen. Der Schwerpunkt der **Interaktion** liegt in diesem Zusammenhang hingegen im Dialog zwischen Benutzer und System bei der Datenanalyse. Der hier verwendete Begriff stammt aus der Human-Computer-Interaction¹⁶ (HCI) als Teilgebiet der Informatik. Das HCI widmet sich der benutzergerechten Gestaltung von interaktiven Systemen und den Schnittstellen zwischen Mensch und Maschine.

¹⁶ dt. Mensch-Computer-Interaktion

Eine genaue Begriffsabgrenzung ist schwierig. Es wird jedoch im Bereich der HCI nach [SJ07, Marcus S. 243]¹⁷ von

„den Eingabemitteln mit denen der Benutzer mit einem System kommuniziert, als auch die vom System erhaltene Rückmeldung“

gesprochen. Hierbei sind jegliche Arten von Eingabemethoden, wie herkömmliche, serielle Eingabegeräte oder komplexe Objekt-, Bewegungs- oder Spracherkennung gemeint. Die Systemrückmeldung umfasst außerdem alle wahrnehmbaren Antworten, wie visuelle, auditive, taktile Signale und deren Kombinationen. Die dabei ablaufenden Kommunikationsprozesse werden im Interaktionsdesign beschrieben und nehmen Bezug auf Orientierung, Navigation, Gestaltung der grafischen Benutzungsoberfläche (GUI) und Systemreaktion.

Allgemein werden somit die Möglichkeiten des Benutzers hinsichtlich Wahrnehmung und Einflussnahme während der Präsentation beschrieben. Dabei wird ein System als **interaktiv** bezeichnet, wenn es fortwährende Benutzereingaben erlaubt und die Systemantwort so schnell erfolgt, dass der Benutzer diese in seinem Arbeitsablauf typischerweise abwartet, ohne andere Arbeiten auszuführen. Wird dem Benutzer keine Interaktion zur Verfügung gestellt, wird dieser als reiner Betrachter der präsentierten Darstellung angesehen. Von diesem Fall abgesehen, können verschiedene Ausprägungen der Interaktion nach [VDI03] unterschieden werden:

Navigation in der Präsentation erlaubt dem Benutzer, die Präsentation über bestimmte Befehle, wie „Start“, „Stop“, „Springe zu“ oder „Änderung des Bildausschnittes“, zu steuern.

Interaktion mit dem grafischen Modell gestattet die Veränderung der Visualisierung anhand von Parametern und ermöglicht die Manipulation von zum Beispiel Blickrichtung, Standort, Ausschnitt der virtuellen Kamera.

Interaktion mit dem Datenmodell ist die teilweise oder grundlegende Einflussnahme auf die Visualisierung durch Veränderung der Basisdaten.

Interaktion in und mit dem Modell (Immersion) bezeichnet die gegenseitige Einflussnahme von Benutzer und Modell. Der Benutzer taucht dabei vollständig in eine „Virtuelle Welt“ ein und wird Teil von dieser. Zur Abschirmung von Außeneinflüssen und um den Wahrnehmungseindruck zu steigern, werden oft spezielle Eingabe- und Ausgabegeräte eingesetzt.

Ziel der Interaktion ist es, den Anwender mit Hilfe von geeigneten Interaktionstechniken in die Lage zu versetzen, Eigenschaften von untersuchten Objekten zu identifizieren und effizienter an gewünschte Informationen zu gelangen. Neben den bei der Visualisierung ebenfalls zu berücksichtigenden Attributen, wie möglichst selbsterklärende Farben,

¹⁷ „... the means by which users communicate input to the system, as well as the feedback supplied by the system.“

Formen und Größen, werden auch Navigationselemente und Benutzerführung betrachtet. Diese sollten ohne großen Lernaufwand verständlich und intuitiv bedienbar sein. Weiterhin ist der Benutzer bei der Navigation durch die Visualisierung, Menüführung, Erledigung spezieller Aufgaben und der Orientierung im System zu unterstützen.

Bei der Applikationsentwicklung werden häufig spezielle Interaktionen implementiert, die grundsätzlich ähnliche Aktionen repräsentieren, aber nicht direkt übertragbar sind. Systemspezifische Adaption, als auch Berücksichtigung der Nutzerintention, sind essentielle Bestandteile bei der Interaktionskonzeption, jedoch wird dadurch die Definition eines einheitlichen Aktionsschemas zu einer großen Herausforderung. In [YKSJ07] wurde durch die Betrachtung von Informationsvisualisierungssystemen der Versuch unternommen, eine solche **Taxonomie**¹⁸ für die Interaktion aufzustellen. Dabei entstanden folgende sieben Hauptkategorien:

Selektieren bezieht sich auf die Markierung interessanter Elemente. Der Benutzer wird in die Lage versetzt, bestimmte Elemente gesondert zu kennzeichnen, um diese auch bei Änderungen in der Präsentation schnell von anderen unterscheiden zu können. Dies ist insbesondere von Vorteil, wenn viele ähnliche oder eine große Anzahl von Elementen gleichzeitig dargestellt werden. Weiterhin kann der Benutzer durch Selektion eine Auswahl treffen und mit dieser anschließend weitere Aktionen, zum Beispiel unabhängig von der Gesamtmenge, durchführen. So wird das Selektieren meist mit weiteren Aktionen verbunden, um die Nutzerinteraktion zu unterstützen.

Erkunden beschreibt die Veränderung der Darstellung, so dass andere Teilmengen des Datensatzes betrachten werden können. Je nach Größe eines Datensatzes ist es fallweise nicht möglich, zum Beispiel durch Begrenzung der Darstellungsfläche oder Beschränkungen der kognitiven Fähigkeiten des Menschen, die gesamte Menge an Elementen gleichzeitig darstell- und wahrnehmbar zu machen. Nach Betrachtung einer Teilmenge, kann der Benutzer zu anderen Teilen wechseln, um weitere Auswertungen vorzunehmen. Dabei kommt es meist nicht zu abrupten, vollständigen Änderungen, sondern es wird ein fließender Übergang zwischen hinzukommenden und entfernten Elementen realisiert. Beispiele hierfür sind das freie Verschieben eines Darstellungsausschnitts (Panning), die Bewegung des Darstellungsfokus von Punkt A nach Punkt B (Direct-Walk) oder der Sprung zu einer anderen Stelle mittels eines Querverweises (Hyperlink).

Rekonfigurieren verweist auf die Präsentation der Elemente in einer anderen Anordnung. Durch die räumliche Umgestaltung des Darstellungslayouts können unterschiedliche Perspektiven auf einen Datensatz verwirklicht werden. Dies ist besonders von Nutzen, um verborgene Charakteristika der Daten ersichtlich zu machen, welche in einer einzelnen Anordnung kaum hinreichend zu veranschaulichen sind. Ein Beispiel hierfür ist die Sortierung der Elemente anhand von Datenwerten oder

¹⁸ Eine Taxonomie oder ein Klassifikationsschema ist ein einheitliches Verfahren oder Modell, um Objekte bezüglich eines Kontextes in Kategorien oder Klassen einzuordnen.

auch komplexere Darstellungsänderungen, um Verdeckung und Überlappung bei Elementanhäufungen zu vermeiden.

Kodieren ist die Einflussnahme auf die grundlegende visuelle Präsentationsform und die Modifikation der Elementdarstellung (Farbe, Form, Größe, etc.). Durch Veränderung visueller Komponenten kann direkt Einfluss auf das Verständnis und die präattentive Wahrnehmung des Nutzers genommen werden. So können zum Beispiel durch Farbkodierung besondere Beziehungen, Klassifikation oder Verteilung von Elementen hervorgehoben werden. Wird außerdem die grundlegende Darstellungsart geändert, wie der Übergang von einem Kreisdiagramm (Pie-Chart) zu einem Histogramm, werden neue Aspekte des Datensets sichtbar. Dynamische Regulation der Darstellungsparameter und die Visualisierung von Übergängen bzw. Zusammenhängen verschiedener Darstellungsformen helfen dem Benutzer, ein prädestiniertes Kodierungsschema zu finden.

Abstrahieren/Verfeinern bedeutet die Anpassung des dargestellten Detaillierungsgrades. Der Benutzer kann Einfluss auf die Abstraktion der Elemente nehmen und so das Darstellungslevel von einer Übersicht bis hin zu individuellen Daten konkretisieren oder generalisieren. Je nach der Intention ist so die Untersuchung der Daten auf kontextueller Ebene oder auch von spezifischen Attributen möglich. Exemplarisch hierfür sind Details-On-Demand-Techniken¹⁹, wie zum Beispiel die Anzeige von zusätzlichen Informationen (Tooltips) bei der Aktivierung von Elementen. Ein weiteres Konzept ist die Skalierung der Darstellung (Zoom), wobei die eigentliche Präsentation im Wesentlichen erhalten bleibt. Durch die Veränderung des Maßstabs kommen hierbei Details klarer zum Vorschein oder gehen in den Kontext über.

Filtern ist die bedingte Elementdarstellung nach bestimmten Voraussetzungen. Es werden lediglich Elemente angezeigt, die bestimmte durch den Benutzer definierte Bedingungen erfüllen. Jene Elemente, die nicht den Kriterien entsprechen, werden hingegen ausgeblendet oder differenziert dargestellt. Durch Zurücksetzen oder Anpassung der Filterparameter ist es zu jeder Zeit möglich, ausgeschlossene Elemente wieder in die Darstellung einzubeziehen. Typische Filtermerkmale sind Datengrenzwerte oder klassifizierende Metadaten, die über Schieberegler oder Optionsschaltflächen dynamisch bestimmt werden können. Ein weiteres Beispiel sind Suchfunktionen. Mittels eines Eingabefeldes können individuelle Anfragen gestellt und alle Elemente mit entsprechenden Attributen ermittelt werden.

Verbinden bezieht sich auf die Anzeige von im Zusammenhang stehenden Elementen. Verbinden wird genutzt, um Zusammenhänge und Beziehungen zu bereits angezeigten Elementen hervorzuheben oder relevante ausgeblendete Elemente anzudeuten. Dies ist insbesondere bei der gleichzeitigen Darstellung von verschiedenen Präsentationsformen hilfreich. Beispielsweise können so die gleichen Elemente in unterschiedlichen Diagrammartentypen hervorgehoben werden, um deren

¹⁹ dt. Details-Auf-Abruf-Techniken

Eigenschaften zu analysieren. Des Weiteren sind Verbinden-Funktionen auch in einzelnen Darstellungen sinnvoll, wie zum Beispiel die Markierung aller zu einer Gruppe gehörigen Elemente.

Ergänzend sind weitere Aktionen, wie das Vor-, Rück- und Zurücksetzen von Ständen einer Aktionshistorie oder Änderung von Einstellungsparametern der Anwendung, möglich. Diese Operationen haben allerdings eine allgemeine Charakteristik und sind nicht spezifisch für die Informationsvisualisierung.

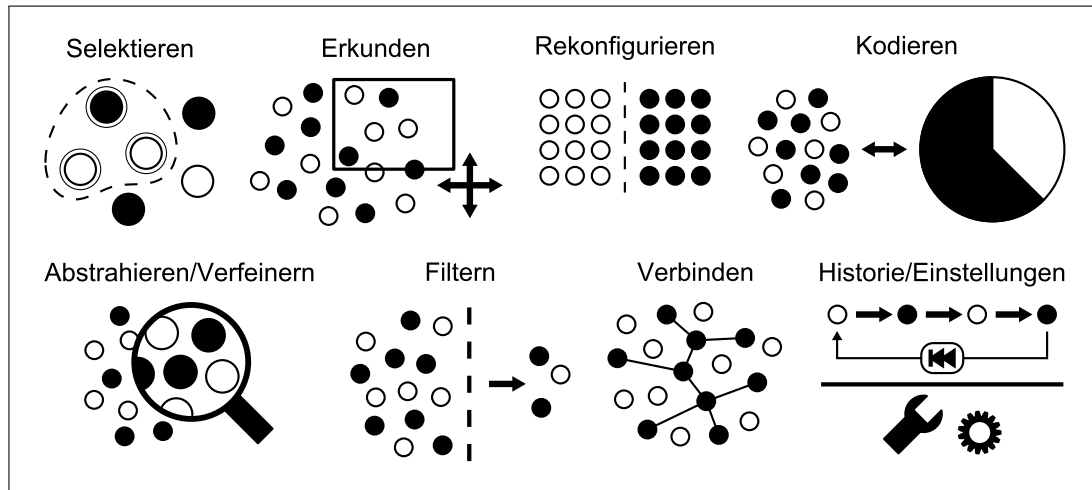


Abbildung 2.14: Grundlegende Interaktionsarten der Informationsvisualisierung

Eines der wichtigsten Interaktionskomponenten bei der Bedienung beeinflussbarer Visualisierungssysteme ist die Navigation mit und durch die präsentierten Daten. **Navigation**²⁰ bezeichnet die Suche nach Informationen in einer Datenmenge, durch das Verfolgen von Beziehungsstrukturen. Allgemein wird bei hierarchischen Daten auch von Drill-down²¹ gesprochen. Dabei werden die speziellen Eigenschaften und Beziehungen der Datenelemente genutzt, um die Analyse bzw. Suche nach konkreten Informationen schrittweise zu verfeinern. Das Gegenstück bilden Drill-up Methoden, die wiederum die inkrementelle Generalisierung der Daten, also ein „Aufsteigen“ in der Hierarchie, realisieren. Diese Funktionen bilden, neben dem GUI-Design, einen wichtigen Teil der Softwareergonomie von interaktiven Systemen mit dem Zweck der Erkenntnisgewinnung aus Datenbeständen und tragen wesentlich zu deren Gebrauchstauglichkeit bei.

Der bedeutsamste Anspruch an die Benutzung eines interaktiven Systems stellt die Vermeidung des **Orientierungsverlusts** dar. Ist es dem Benutzer nicht möglich, sich angemessen zu orientieren, hat dies meist negative Auswirkungen auf den Arbeitsablauf. So ist zum Beispiel nicht nachvollziehbar, wie ein bestimmter Zustand erzeugt wurde oder wie ein anderer Zustand hergestellt werden kann. Ferner ist es ausgeschlossen, weitere relevante Navigationsschritte zu identifizieren, Informationen in Zusammenhang und Kontext zu betrachten und bereits analysierte Elemente zu differenzieren. Dieses Problem tritt

²⁰ Auch Browsing (engl. browse around: sich umsehen, stöbern)

²¹ dt. aufreißen, hinein graben

auf, wenn der Benutzer nicht mit dem zur Verfügung stehenden Aktionsschema vertraut ist bzw. ihm nur ungenügende Informationen und Hilfestellungen für die Bedienung geboten werden. Benutzerfreundliche Systeme dokumentieren in diesem Zusammenhang die Arbeitsschritte der Benutzer und stellen eine Aktionshistorie abrufbar zur Verfügung. Weiterhin werden etwaige Fehleingaben grundsätzlich ausgeschlossen oder dem Benutzer gegebenenfalls kenntlich gemacht. Eine weitere Ursache für die Problematik des Orientierungsverlusts, ist die kognitive Überlastung bei der Interaktion und Analysetätigkeit. Gründe hierfür sind beispielsweise die Überladung von GUI-Komponenten und die gleichzeitige Darstellung von zu vielen Elementen und Beziehungen. Die dargestellten Elemente besitzen häufig unzureichende Unterscheidungsmerkmale und bieten dem Benutzer zu wenig Anhaltspunkte für die Orientierung. Die Berücksichtigung dieser Problematik während der Visualisierungs- und Interaktionskonzeption und der Einbezug geeigneter Funktionen und Bestandteile, können in diesem Zusammenhang zur Vermeidung des Orientierungsverlustes beitragen.

Wie schon in diesem Kapitel angedeutet, sind Visualisierung und Interaktion als zusammenhängende Teile zu betrachten und können nicht gegenseitig ausgeschlossen werden. Interaktionen führen zur Aktivierung von Veränderungen in der Datenrepräsentation, welche wiederum Auswirkungen auf die Darstellung haben. Ohne Interaktion beschränkt sich das Potenzial eines Visualisierungssystems auf statische Bilder oder autonome Animationen. Unter Berücksichtigung der Datensatzmenge wird die Anwendbarkeit von reinen Visualisierungen gleichzeitig stark limitiert. Tatsächlich kann die Interaktion nie vollständig ausgeschlossen werden. Beispielsweise findet bei der Betrachtung von statischen Informationen, wie einem gedruckten Postern, grundsätzlich eine Art passive Interaktion statt (Rotation des Posters, Betrachtung von fern und nah, Notizen, etc.). Die Berücksichtigung grundlegender Interaktionsmöglichkeiten bereits bei der Visualisierungskonzeption dient der Vermeidung von späteren Problemen und Grenzen. Durch effektive Verbindung von Interaktion und Visualisierung wird die Wahrnehmung des Benutzers gefördert und die Analysearbeit unterstützt.

2.6 Klassifikation von Visualisierungs- und Interaktionskonzepten

Eine allgemeingültige **Klassifikation** von Visualisierungs- und Interaktionskonzepten ist eine nur bedingt lösbare Aufgabe. Je nach Betrachtungsweise sind hierfür unterschiedliche Randbedingungen ausschlaggebend. Dazu zählen die Variablendimension und der Charakter (statisch, dynamisch) der zugrunde liegenden Daten, die abgezielte räumliche Darstellung (1D, 2D, 3D oder hybride) oder auch die Datenaspekte. Diese erfordern häufig spezielle Anpassungen, um eine sinnvolle Darstellung zu realisieren. Zugleich sind diese Eigenschaften wichtige Kriterien für die Auswahl von geeigneten Visualisierungskonzepten. Des Weiteren lassen sich konkrete Konzepte nicht eindeutig zuordnen, da sie Hybriden einzelner Klassen darstellen. Dies liegt darin begründet,

dass Nachteile oder Grenzen einer Visualisierungsform ausgeglichen werden müssen oder auch durch die Interaktion dynamische und veränderbare Visualisierungsmodelle geschaffen werden. Dabei ist es auch möglich, Visualisierungsarten zu kombinieren oder ineinander zu überführen. Aus diesem Grund ist daher bei der Diskussion, entsprechend den elementaren Ausprägungen der Produktstrukturen (insbesondere Aggregation und Assoziation), von zugrundeliegenden hierarchischen bzw. netzartigen Daten und Visualisierungskonzepten auszugehen. Für diese wird folgend eine Einteilung geschaffen und deren Grenzen, Eigenschaften und Besonderheiten werden hervorgehoben. Darüber hinaus sind hierbei auch unterschiedliche Komplexitätsstufen und Ausmaße der Datenstrukturen zu betrachten, um deren Eignung und Adaption zu erörtern.

Hauptmerkmal **hierarchischer und netzartiger Strukturen** ist deren grundlegendes Ordnungssystem von Aufbauschemen bzw. Zugehörigkeiten und Beziehungen. Neben weiteren Attributen sind diese Eigenschaften eines der wichtigsten Kriterien bei der visuellen Beschreibung der Daten und meist ausschlaggebend für das entstehende Layout. Unter Berücksichtigung dieser, wird die Einteilung der Visualisierungsmodelle in die allgemeinen Klassen Kapselung (Enclosure) und Knoten-Kanten (Node-Link) vorgenommen. Die Klassen umfassen wiederum verschiedenartige Ausprägungen mit unterschiedlichen Merkmalen und Attributen. Weiterhin existieren spezielle Modelle, die auf konkrete Anwendungen oder Daten²² abgestimmt und nur schwer zu verallgemeinern sind. Diese werden hier als Sonderfall betrachtet. Grundsätzlich sind die Grenzen der Kategorien jedoch fließend und die eindeutige Zuordnung von bestimmten Ausprägungen ist teils nur bedingt möglich.

Kapselung oder auch Treemap ist eine Methode für die Abbildung von vorrangig hierarchischen Daten und wurde in seiner ursprünglichen Form 1991 erstmalig von Ben Schneidermann in [Shn91] vorgestellt. Es handelt sich dabei um einen raumbeschränkten und -füllenden Ansatz zur Visualisierung von Baumstrukturen, wobei Elemente mittels begrenzter Flächen repräsentiert werden. Einem Ausgangselement wird eine Fläche zugewiesen und diese entsprechend der Anzahl der untergeordneten Elemente unterteilt. Die neu entstandenen Regionen werden den Subelementen zugeordnet und der Vorgang wird, der Hierarchie folgend, rekursiv für alle Elemente der Struktur gleichermaßen fortgeführt. Die Struktur der Daten wird somit anhand von verschachtelten Formen repräsentiert, die sich jeweils innerhalb übergeordneter Formen befinden und von diesen einen Teilbereich einnehmen. Als Resultat können auf begrenztem Raum auch sehr große und komplexe Strukturen dargestellt werden. Für dieses Konzept existieren vielfältige Ausführungen, deren gemeinsame Charakteristika in Tabelle 2.3 zusammengefasst ist.

²² Wissenschaftliche Visualisierungen nutzen zum Beispiel teilweise die physische Form eines Objekts, um darauf erzeugte Informationen aus den zugrundeliegenden Daten abzubilden.

Merkmale	Raumbeschränkter und- füllender Ansatz, bei dem die Datenstruktur verschachtelt dargestellt wird
	Abbildung der Daten durch rekursive Partitionierung des Raumes in Element-repräsentierende Felder
Vorteile	Layout bietet optimal bis zu 100%ige Ausnutzung des Darstellungsraumes
	Komplexe und große Hierarchien können effizient für Überblick und Element-Fokus dargestellt werden
	Effektive Darstellung der Elemente und zugehöriger Attribute und die Eignung für zusätzliche Farb-, Form-, Größenkodierung, etc. der Felder
Nachteile	Topologie der Struktur kann nur erschwert analysiert werden
	Netzartige Strukturen und Querverbindungen von Knoten nur bedingt darstellbar

Tabelle 2.3: Merkmale der Visualisierungskategorie Kapselung

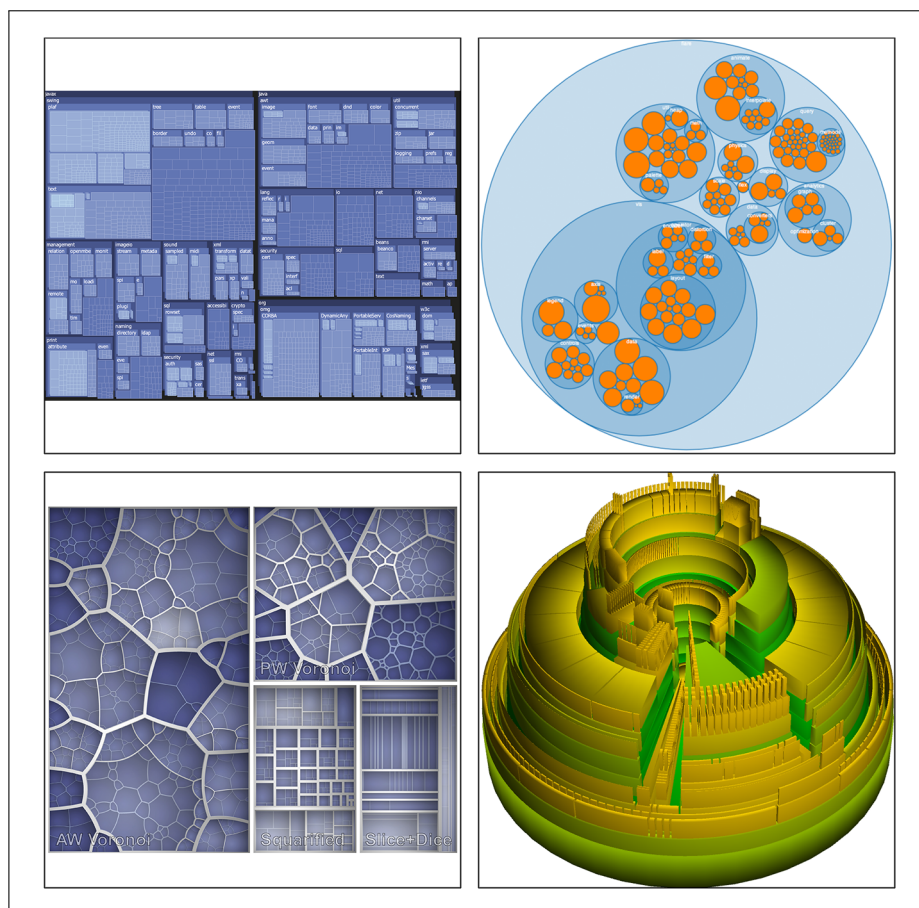


Abbildung 2.15: Beispiele für die Visualisierungskategorie Kapselung (Treemap). Abgebildet sind eine Cascaded Treemap (oben links) [LF08], eine Circular Treemap (oben rechts) [BHO10], Voronoi-, Squarified- und Slice-And-Dice Treemaps (unten links) [BD05] und eine extrudierte 3D-Polar Treemap (unten rechts) [Sch10].

Die **Knoten-Kanten** Klasse umfasst Methoden zur Abbildung von hierarchischen oder netzartigen Strukturen und wird in ihrer allgemeinen Form in der Graphentheorie auch als Graph und für Monohierarchien²³ als Baum bezeichnet. Konventionell werden dabei Elemente abstrakt mit Knoten oder Punkten (Formen, Text, Bilder, etc.) beschrieben und deren Beziehungen als Kanten (Verbindungslinien) illustriert. Bei Bäumen wird die Position der Knoten durch die Strukturtiefe bzw. -ebenen, auf ordinaler Y-Achse und Abgrenzung zu Geschwisterknoten²⁴, auf nominaler X-Achse, bestimmt. Von einem Knoten ausgehende Kanten repräsentieren hierbei entsprechend die Zugehörigkeit zum übergeordneten Element bzw. Beziehung zu direkt untergeordneten Elementen. Bei Graphen hingegen können weitere Kanten (Querverbindungen) existieren, welche zusätzliche Zusammenhänge zwischen den Knoten symbolisieren. Für dieses Konzept existieren verschiedenartige Ausführungen, deren gemeinsame Merkmale in Tabelle 2.4 aufgeführt sind.

Merkmale	Räumlich ausbreitender Ansatz, der Struktur als Knoten (Elemente) und Kanten (Beziehungen) repräsentiert
	Vielfältige Darstellungsformen durch facettenreiche Layout-Techniken
Vorteile	Verständliche und effektive Übersicht der Struktur-Topologie
	Beziehungen können explizit und durch verschiedenartige Kodierung veranschaulicht werden
	Diverse Zusammenhänge können durch zusätzliche Kanten (Querverbindungen) realisiert werden
Nachteile	Spezielle Strukturarten können zu extremen Proportionen bzw. Seitenverhältnissen des Layout führen
	Strukturen beinhalten einen wesentlichen Anteil an Leerraum (ca. 50%)
	Komplexe Strukturen können eine Vielzahl von Knoten und Verbindungslinien enthalten, wodurch Verständnis und Übersichtlichkeit des Layout beeinträchtigt werden kann (zum Beispiel durch Linienüberschneidung)

Tabelle 2.4: Merkmale der Visualisierungskategorie Knoten-Kanten

Die räumliche **Dimensionalität** – Freiheitsgrade des visualisierten Informationsraums – der Visualisierungsmethoden lässt Klassifizierungen der verschiedenen Ausprägungen zu. Eindimensionale Darstellungen sind streng genommen nicht möglich. Gemeint sind hier jedoch vor allem zeilenweise präsentierte Darstellungsformen, die sich auf sequenzielle Daten, wie Textdokumente, Quellcode oder Listen beziehen und nicht zwingend eine zweite Dimension erfordern. Beispiele für eindimensionale Visualisierung sind zum Beispiel Balken- und Strahlendiagramme. 2D-Visualisierungen bezeichnen hingegen digitale Bilder, Grafiken, Diagramme oder visuelle Präsentationen, wie Übersichtskarten und Matrixdarstellungen. Hierzu zählen beispielsweise auch die entsprechenden Methoden aus Abbildung 2.15 und 2.16. 3D-Konzepte können durch den zusätzlichen Freiheitsgrad physische Objekte modellieren bzw. mehr Informationen kodieren, welche andernfalls

²³ Bei Monohierarchien ist jedem Element höchstens ein anderes Element unmittelbar übergeordnet.

²⁴ Geschwisterknoten sind Elemente der gleichen Klasse bzw. sind dem gleichen Element direkt untergeordnet.

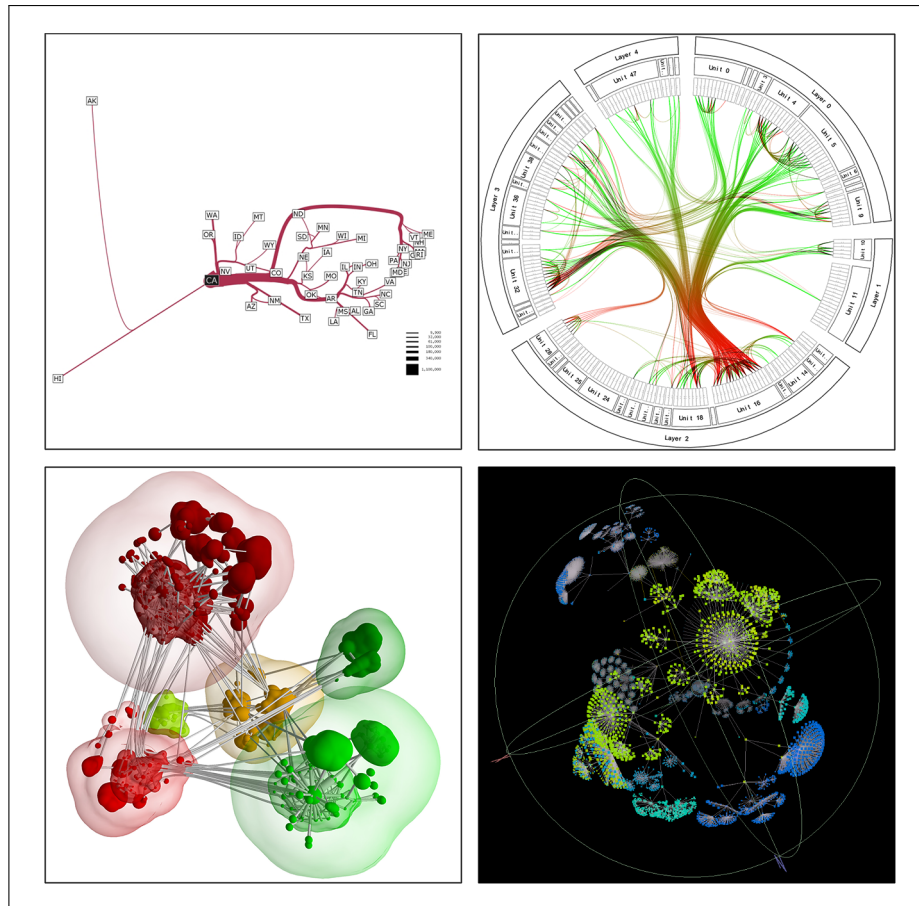


Abbildung 2.16: Beispiele für die Visualisierungskategorie Knoten-Kanten. Abgebildet sind eine Flow Map (oben links) [PXY05], Hierarchical Edge Bundles (oben rechts) [Hol06], ein Clustered Graph Layout (unten links) [BD07] und ein hyperbolisches 3D-Layout (unten rechts) [Wal05].

weggelassen oder anderweitig aufbereitet werden müssten. Umgesetzt werden diese überwiegend durch Abbildung der Informationen auf 3D-Geometrien, Projektion von 2D-Darstellungen auf 3D-Flächen oder anhand perspektivischer Verzerrungen, um räumliche Eindrücke zu erzeugen. In Abhängigkeit vom Anwendungstyp ist die Benutzerfreundlichkeit von 3D-Schnittstellen – 3D-Navigation mittels 2D-Eingabegeräten (Maus, Tastatur) – allerdings umstritten. Es gibt fortlaufende Untersuchungen, ob das räumliche Gedächtnis des Menschen durch 3D-Darstellung und -Interaktion besser genutzt und somit die Informationswahrnehmung gesteigert werden kann [Coc04]. Gegenüber 2D profitieren 3D-Konzepte zwar von Tiefenwahrnehmung und höherer Informationsdichte, jedoch kann die statische Darstellung und Verständlichkeit auch durch Einschränkungen des Betrachtungswinkels oder Verdeckungsprobleme beeinträchtigt werden [ET08]. Alternativ existiert die Erweiterung von herkömmlichen 2D-Konzepten zu 2½D bzw. Pseudo-3D Präsentationen. Zur Unterstützung des Interpretations- und Erinnerungsvermögens werden hierbei beispielsweise Glanzlichter, Schattierungen von Elementen oder Isometrie, Dimetrie und Kavalierperspektive²⁵ eingesetzt. Letztlich ist die Abhängigkeit und Auswahl der räumlichen Dimension der Visualisierung anwendungsspezifisch und vorrangig im Zusammenhang mit den zugrundeliegenden Daten zu betrachten.

Die **Relation zwischen Visualisierungen** beschreibt die übergeordnet wechselseitige Beziehung zwischen unterschiedlichen Präsentationsformen gleicher oder ähnlicher Bezugsdaten. Dabei können zum Beispiel spezielle Muster in den einzelnen Darstellungen verglichen und der Bezug von Elementen in alternativen Informationskodierungen untersucht werden [CC07]. Neben dem direkten Vergleich diverser Präsentationsformen (z.B. Knoten-Kanten und Kreisdiagramm), sind einzelne Zustände von zum Beispiel zeitabhängigen Daten darstell- und gegenüberstellbar. Entwicklungen und Abhängigkeiten von Einflussfaktoren können so bei dynamischen Abfolgen und Versionen nachverfolgt werden.

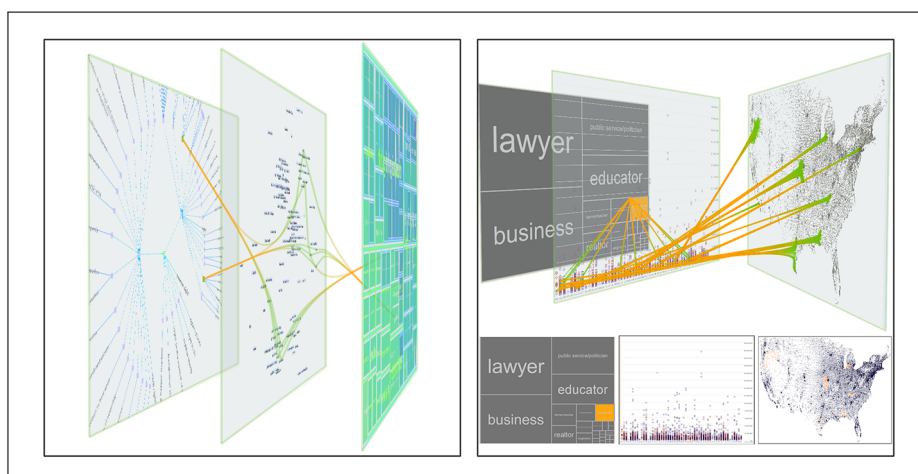


Abbildung 2.17: Beispiele für Relationen zwischen Visualisierungen nach [CC07]

²⁵ Isometrie, Dimetrie und Kavalierperspektive sind Begriffe aus der Darstellenden Geometrie und bezeichnen verschiedene perspektivische Ansichten auf räumliche Geometrien.

Die **technisch-wissenschaftliche Visualisierung** beschäftigt sich mit Darstellung dreidimensionaler Phänomene, wie Architektur, Meteorologie oder auch Biologie. Der Schwerpunkt liegt auf dem realistischen Rendering von Volumen, Oberflächen, Lichtquellen und fallweise deren zeitlicher Veränderung. Dabei können komplexe physikalische Prozesse verständlich abgebildet und die Eigenschaften und Besonderheiten geometrischer Modelle analysiert werden. Gegenüber der Informationsvisualisierung, die sich hauptsächlich auf abstrakte Daten bezieht, wird bei der technisch-wissenschaftlichen Visualisierung die Interpretationsqualität von realitätsnahen 3D-Modellen genutzt. Im Ingenieurwesen und der angewandten Wissenschaft werden vor allem 3D-Geometriemodelle eingesetzt, die mittels eines 3D-CAD-Systems²⁶ nach diversen Modellierungsmethoden entstehen. Diese Modelle dienen der virtuellen Analyse der Gebrauchstauglichkeit unter speziellen Bedingungen oder auch Soll-Ist-Vergleichen in der Instandhaltung und anderen MRO-Prozessen²⁷.

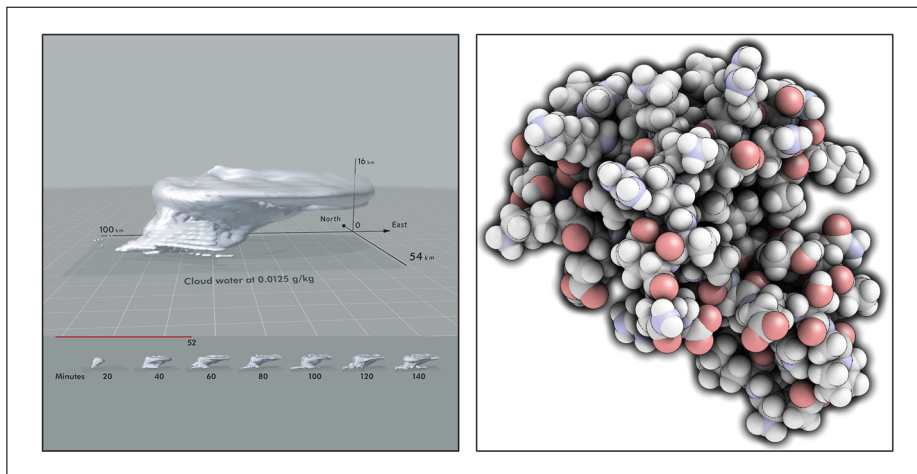


Abbildung 2.18: Beispiele für technisch-wissenschaftliche Visualisierung. Abgebildet ist die Visualisierung eines Gewittersturms (links) nach [BB95] und die Darstellung eines Moleküls (rechts) nach [TCM06].

2.7 Benutzungsschnittstellen und Präsentation

Eine **Benutzungsschnittstelle** (auch Mensch-Maschine-Schnittstelle) ist ein wichtiger Bestandteil eines Mensch-Maschine-Systems und fungiert als Verbindungsglied bei der Kommunikation, dem Austausch von Informationen und der Interaktion zwischen Benutzer und System. Nach [DIN08] umfasst eine Benutzungsschnittstelle somit:

„alle Bestandteile eines interaktiven Systems (Software oder Hardware), die Informationen und Steuerelemente zur Verfügung stellen, die für den Benutzer notwendig sind, um eine bestimmte Arbeitsaufgabe mit dem interaktiven System zu erledigen.“

²⁶ 3D-Systeme für rechnergestützten Entwurf und Konstruktion.

²⁷ dt. Wartung, Reparatur und Instandhaltung

Bei der Schnittstellengestaltung steht die Nutzbarkeit im Vordergrund, wobei Bedürfnisse des Menschen als Bediener und Beobachter und Anforderungen des Systems zu berücksichtigen sind. Softwaretechnisch erfolgt die Bereitstellung der Informationen über Visualisierungssysteme und die Bedienung wird anhand visueller Komponenten einer grafischen Benutzungsoberfläche realisiert.

Das **Präsentationsproblem** beschreibt in der Informationsvisualisierung die Problematik, dass darzustellende Informationsräume oftmals wesentlich größer als der zur Verfügung stehende Anzeigebereich sind. Dies resultiert vor allem aus einer geforderten überdimensionalen Informationsmenge und -dichte. Zum Beispiel ist so die gleichzeitige Anzeige von mehreren Sichtweisen, Fallunterscheidungen und Attribut- und Variablenwerten stark limitiert. Des Weiteren stellen die Lokalisierung festgelegter Elemente, die Analyse und Interpretation von Details und der Bezug zur unmittelbaren Informationsumgebung dadurch eine hohe Beanspruchung für den Benutzer dar. Verschiedene Ansätze, wie Datenkodierung, Interaktion, Darstellungstransformation und Mischformen beschäftigen sich mit der Maximierung der abgebildeten Elemente. Ein weiterer Bestandteil ist die Optimierung des Darstellungsbereiches. Der gebräuchlichste Lösungsansatz ist der Einsatz von **Bildlauf** (Scrolling), der das Verschieben des Bildschirminhalts bzw. des Anzeigeausschnitts bezeichnet. Zur Bedienung dienen dabei Schieber und Bildlaufleisten, mit denen der Anzeigeeinhalt in verschiedenen Geschwindigkeiten verändert werden kann. Nachteilig ist die limitierte, lineare horizontale und vertikale Navigation und die oft nicht existente Vorschau von Informationen außerhalb des Anzeigebereichs.

Alternativ dazu sind **skalierbare Benutzungsoberflächen (ZUI)**²⁸, die zusätzlich Vergrößerung und Verkleinerung (Zoomen) von einzelnen Elementen oder der gesamten Darstellung ermöglichen. Der Benutzer ist in der Lage, über bestimmte Eingaben den Skalierungsfaktor des Darstellungsfeldes zu steuern und ähnlich zum Bildlauf den Anzeigeausschnitt zu verschieben (Panning). Durch die nicht-lineare Navigation wächst das Potenzial von ZUIs mit der Größe des Informationsraums, wobei die Anordnung der Darstellungselemente räumlich und maßstäblich zu organisieren sind. Neben der klassischen geometrischen Skalierung existieren weitere Arten, wie zum Beispiel der semantische oder zielgerichtete Zoom²⁹. Das Hauptproblem bei der Arbeit mit ZUIs stellt für den Benutzer der ungenügende Kontext dar. Durch fortwährendes Zoomen und Verschieben des Anzeigeeinhalts gehen gleichzeitig kontinuierlich Anhaltspunkte für die Orientierung im Informationsraum verloren. In [JF98] mit dem Titel „*Kritische Zonen im Distanznebel*“³⁰ werden außerdem weitere schwerwiegende Komplikationen bei großen, infiniten und mehrskaligen Informationsräumen vorgestellt. Darin wird die Uneindeutigkeit beim Zoomen in einen Leerraum zwischen den Elementen beschrieben. In dieser

²⁸ engl. Zoomable User Interface (ZUI)

²⁹ Der semantische Zoom (Semantic Zoom) ändert das Erscheinungsbild der Objekte entsprechend dem Skalierungsfaktor. Beim zielgerichteten Zoom (Goal-Directed Zoom) hingegen wählt der Benutzer eine Repräsentationsform des Objektes und Skalierungs- und Transformationsänderungen werden automatisch vom System ausgeführt.

³⁰ Critical Zones in Desert Fog – Aids to Multiscale Navigation

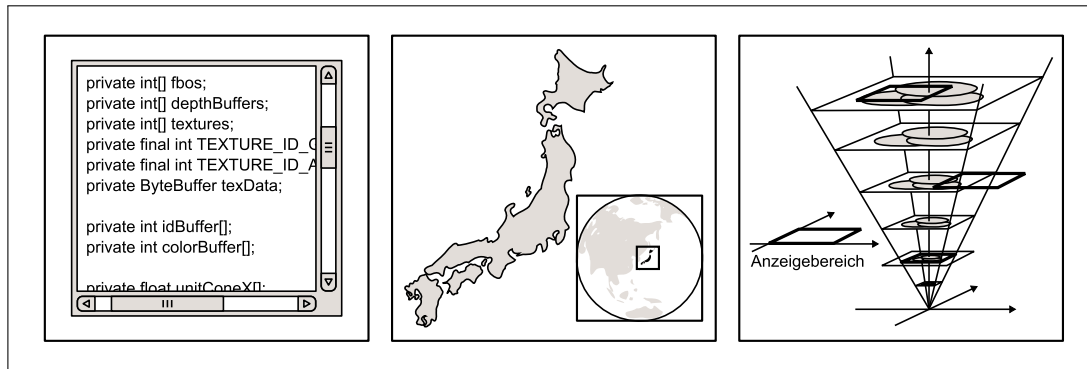


Abbildung 2.19: Schematische Beispiele für verschiedene Arten von Benutzungsoberflächen. Abgebildet sind Interfaces für Bildlauf (links), Overview plus Detail (mitte) und das Prinzip einer ZUI nach [FB95].

Situation hat der Benutzer keine Gewissheit darüber, ob sich weitere Informationen in dieser Richtung befinden oder inwieweit durch weiterführendes Zoomen neue Elemente zum Vorschein kommen.

Ein weiteres Konzept für Benutzungsschnittstellen sind **Übersicht plus Detail Interfaces (O+d)**³¹, für die ein Layout mit mehreren Informationsbereichen bzw. -fenstern charakteristisch ist. Die Fenster unterscheiden sich durch die jeweilige Sichtweise, Übersicht und das Detail auf den Informationsraum und besitzen oft gegenseitige Orientierungspunkte. So wird das Sichtfeld des Detailfensters häufig als Bildfeldbox anhand eines Rechtecks in der Übersichtsanzeige dargestellt, wodurch der Benutzer sich im Kontext des Informationsraums orientieren kann. Durch Interaktion mit der Übersicht kann der Benutzer zudem die Detailsicht direkt an einen anderen Punkt versetzen. Dabei ist er jedoch gezwungen, permanent visuell und mental zwischen den Fenstern umzuschalten. In [HBP02] zeigt eine Studie zum Vergleich von ZUIs mit und ohne Übersicht, dass die Teilnehmer im Allgemeinen mit reinen Detailansicht schneller sind, jedoch mehrheitlich eine zusätzliche Übersicht bevorzugen.

Fokus und Detail-In-Kontext Techniken komprimieren wiederum Ausschnitte einer Darstellung mit dem Ziel der Informationsanreicherung und der Veranschaulichung von Relationen. Die Methoden heben relevante und aktive Informationen hervor und unterstützen den Benutzer bei der arbeitsbezogenen Orientierung. Ferner werden interessante Informationen mit ihrem Umfeld in Verbindung gebracht, so dass nächstliegende Beziehungen und Verbindungen im Gesamtkontext analysiert werden können. Dies ist von Vorteil, da Übersichten nur eine abstrakte Sicht auf große Informationsmengen erlauben, bei der detaillierte Informationen nur bedingt wahrgenommen werden können. Die Kombination von Übersicht (Kontext) und Detailinformation (Fokus) in einer einzelnen dynamischen Darstellung, ist an das menschliche Sehvermögen angelehnt, wobei besonders wichtige Relationen ersichtlich werden. In konkreten Anwendungen geschieht dies durch Verzerrung der Darstellung um einen Punkt von Interesse [LA94]. Dabei

³¹ engl. Overview plus Detail Interface (O+d)

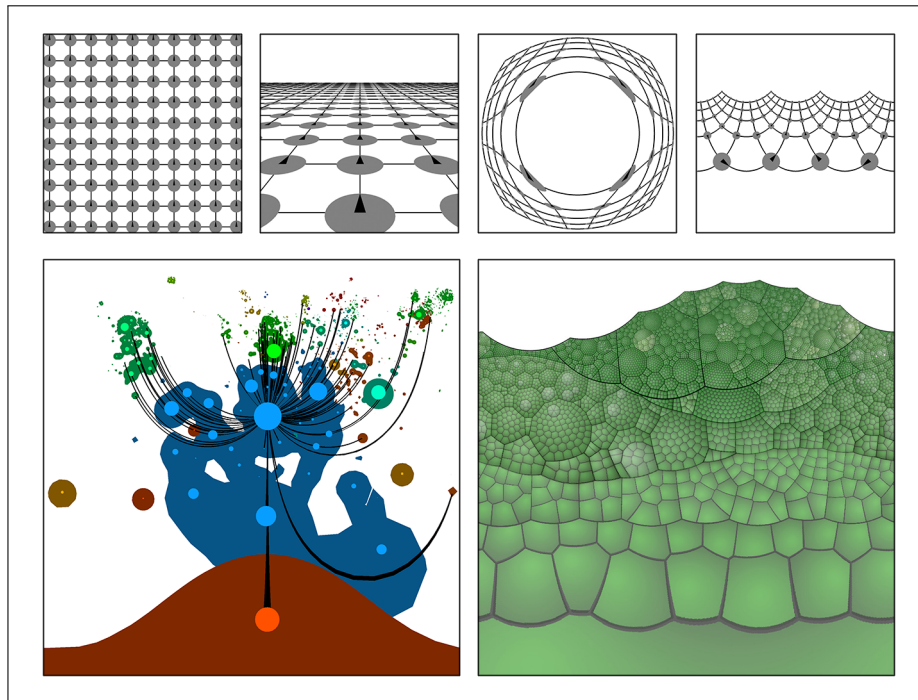


Abbildung 2.20: Beispiele für Fokus und Detail-In-Kontext Techniken nach [BBD06]. In der oberen Bildreihe sind von links nach rechts identische, perspektivische, polyfokale bzw. Fischaugen- und komplex logarithmische Projektionen abgebildet. Die untere Bildreihe zeigt die komplex logarithmische Projektion am Beispiel von Knoten-Kanten- und Kapselung-Layouts.

werden Elemente in der Nähe des Fokus vergrößert und detailreich dargestellt und mit steigender Entfernung kleiner und mit weniger Informationen. Es existieren unterschiedlichste Ausprägungen für die Deformation des Darstellungsraumes, wobei diese sich nicht zwangsläufig auch auf die visuelle Repräsentation der Elemente auswirken muss. Gemein ist diesen Techniken, dass überwiegend durch simple Interaktion³² der Fokus verschoben oder neu gesetzt werden kann. Präzise Navigation ist dabei ohne weitere Anpassungen allerdings problematisch, da sich Elemente durch die Verzerrung mit rapide steigender Geschwindigkeit zum Fokuspunkt bewegen. Die Verzerrung hat außerdem negativen Einfluss auf die Wahrnehmung von Proportionen, Winkeln und Distanzen. Der sinnvolle Einsatz dieser Techniken ist folglich stark anwendungsspezifisch. In Abbildung 2.20 werden einige dieser Techniken vorgestellt.

Im Allgemeinen ist die konkrete Auswahl oder Kombinationen von Schnittstellenkonzepten durch die Visualisierung, die Ausgangsdaten und durch Form und Ausmaß des Präsentationsmediums bedingt. Die Konzeptentscheidung ist darüber hinaus in Abhängigkeit mit der Zielgruppe und den Einsatzfeldern zu treffen.

³² Zum Beispiel Click-And-Drag Aktionen (dt. Klicken und Ziehen) mit einer Maus.

Natural User Interfaces (NUI) oder natürliche Benutzungsschnittstellen ist ein Begriff aus der Entwicklung von Computerschnittstellen und bezieht sich darauf, dass Benutzungsschnittstellen selbst nahezu nicht wahrnehmbar sind bzw. durch sukzessiv erlernte Interaktionen für deren Benutzer unmerklich werden. Durch Abstraktion von Eingabegeräten und -technologie kann der Benutzer sich verstärkt auf die eigentliche Anwendung konzentrieren und es wird das Gefühl von natürlichem Umgang, spontanen Interaktionen und Einklang mit dem Inhalt impliziert. Die Gestaltung der Schnittstelle ermöglicht es, dem Benutzer, mittels minimalem Lernaufwand und intuitiver Bedienung, seine Fähigkeiten in kurzer Zeit zu erweitern und auch komplexe Aktionen zu bewältigen. Durch den Einsatz entsprechender Technologien ist der Benutzer in der Lage, relativ natürliche Zeichen, Bewegungen, Gebärden oder spezielle Operationen und Geräte für die Kommunikation mit der Anwendung einzusetzen. Die Nutzung dieser Techniken, die Entwicklung benutzerfreundlicher Anwendungen und die Gestaltung verständlicher Interaktionen vermitteln dem Benutzer ein intuitives und logisches Gefühl bei der Bedienung. Dies steht teils im Gegensatz zum Einsatz herkömmlicher Eingabegeräte, deren Operationen erst erlernt werden müssen.

Historisch kann NUI als fortführende Phase der Entwicklung vom Kommandozeilen Interface³³ zur grafischen Benutzungsoberfläche verstanden werden. Bei der Kommandozeile werden über die textuelle Eingabe vordefinierter Befehle bestimmte Aktionen ausgeführt, wobei die Kommandos meist limitiert und wenig flexibel sind. Der Übergang zu GUI brachte darauf grafische Bedienkomponenten in den Vordergrund. Diese visuellen Elemente stellen Abstraktionen realer Objekte, wie Arbeitsplatz oder Dokumente dar, und der Benutzer kann direkt mit ihnen interagieren. Das derzeitig dominante Grundkonzept für GUI eines Arbeitsplatzrechners wird häufig auch als WIMP³⁴ bezeichnet, welches die beim Dialog mit dem Benutzer beteiligten Komponenten, wie Fenster, Symbole, Menüs und Zeiger beschreibt. Derartige Kommunikationsmetaphern sind für den Umgang mit NUI Konzepten ohne weitere Anpassungen nur bedingt geeignet. NUI fokussiert die grundlegenden menschlichen Fähigkeiten, wie Berührung, Sehen, Sprechen, Hören, Handschrift, Bewegung und höhere Fertigkeiten, wie Wahrnehmung, Kreativität und Wissbegierde. Ein weiterer Bestandteil ist die Nachbildung realer Milieus und Zustände unter dem Einsatz neuartiger Technologie und Messsysteme, um eine optimierte Wechselwirkung zwischen physischen und digitalen Objekten zu schaffen. Zu diesem Zweck müssen grundlegende GUI Designkonzepte teils überarbeitet und für menschliche Wahrnehmung, Sensorik und Motorik angepasst werden. Aus traditioneller Sicht existieren so unter anderem erhebliche Unterschiede bei der Eingabepräzision verschiedener Interaktionsmethoden, wodurch Adaption grafischer Komponenten (Größe, Position, Aufbau, Eigenschaften) oder gar neue Interaktionsparadigmen erforderlich sind [Saf08]. Ein weiterer Forschungsbereich beschäftigt sich mit dem gleichzeitigen Umgang und dem Informationsaustausch zwischen einem System und mehreren Nutzern. Im Gegensatz zu klassischen Arbeitsstationen, deren Operationen gewöhnlich auf den Dialog mit nur

³³ engl. Command-Line Interface (CLI)

³⁴ WIMP ist ein Akronym für Windows, Icons, Menus, Pointer.

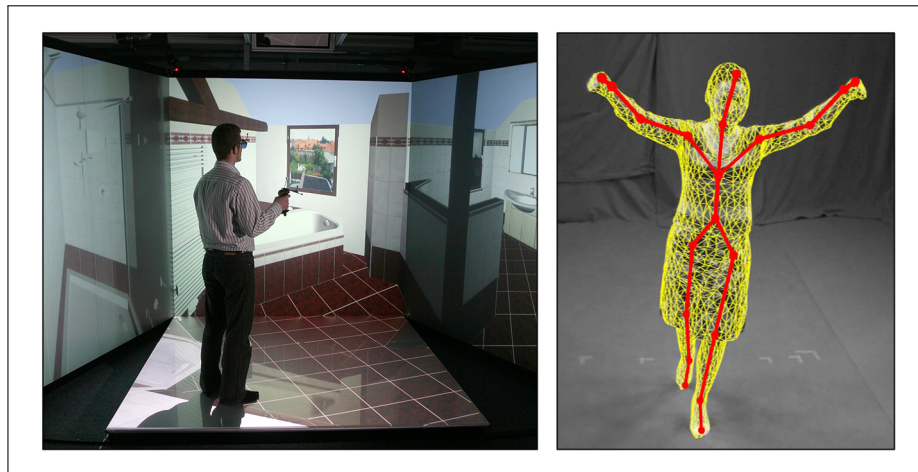


Abbildung 2.21: Beispiele für NUI und Präsentationsmedien. Abgebildet ist die Interaktion mit einer CAVE (links) des VDTC (Quelle: Fraunhofer IFF) und der Systemdialog mittels Skeleton Tracking nach [GSA09].

einem Anwender konzipiert sind, werden dabei Modelle des simultanen kollaborativen und verteilten Arbeitens untersucht. Dies wird vor allem durch die Kombination entsprechender Präsentationsmedien, spezieller Benutzerführung und Eingabemethodik ermöglicht.

Geeignete **Präsentationsmedien und -methodiken** sind oft eng mit der Interaktion verbunden und somit wichtige Bestandteile für das Verständnis und die Wahrnehmung des Nutzers. Das Potenzial der kognitiven Fähigkeiten kann durch neuartige Technologien besser genutzt und somit das Erleben von Kommunikation und dargestellter Information erreicht werden. Hierfür werden bestehende Systeme mit speziellen Komponenten angereichert, virtuelle Realitäten (VR)³⁵ erschaffen oder auch die Realitätswahrnehmung computergestützt erweitert (AR)³⁶. Beim Umgang mit derartigen Systemen wird oftmals auch der Begriff der Immersion genannt, welcher den Zustand des „Eintauchens“ in die künstliche Welt bzw. den intensiven Fokus auf die dargestellten Informationen beschreibt. In der Industrie und anderen Ingenieursdisziplinen werden zu diesem Zweck neben herkömmlichen VR und AR auch Cave Automatic Virtual Environment (CAVE) eingesetzt. Damit sind Unternehmen beispielsweise in der Lage, ihren Kunden einen Eindruck von Produkten in einem bestimmtem Umfeld oder unter gesonderten Anforderungen zu demonstrieren. In Kombination mit entsprechender Interaktion können damit auch komplexe Simulationen oder kreative Konzeptionen in der virtuellen Produktentstehung realisiert werden [Isr09]. Weitere Beispiele sind Head-Mounted Displays, stereoskopische Projektoren oder andersartige 3D-Displays.

Publikationen zu Touchscreens und insbesondere **Multi-Touch Technologien** existieren schon seit geraumer Zeit [LBS85]. Diese finden aber erst seit in den letzten Jahren

³⁵ engl. Virtual Reality (VR)

³⁶ engl. Augmented Reality (AR)

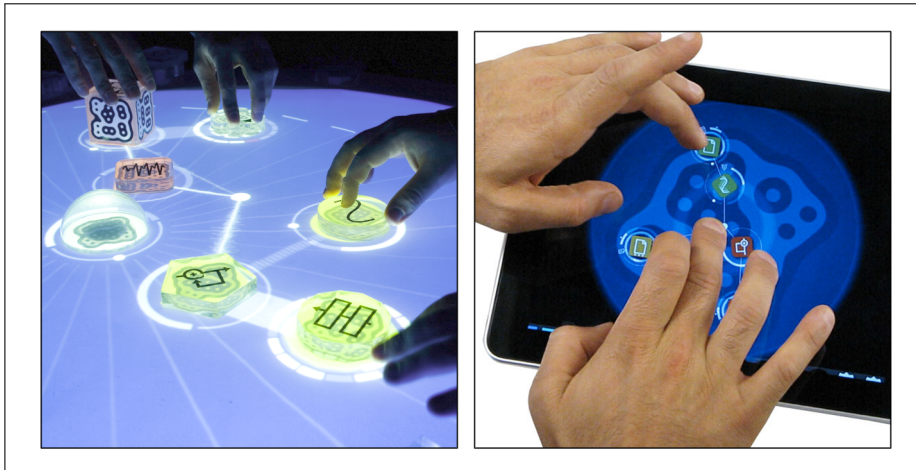


Abbildung 2.22: Beispiele für Multi-Touch Technologien. Abgebildet ist die Interaktion mit einem Multi-Touch System unter Einsatz einer stationären Arbeitsstation (links) und eines mobilen Gerätes (rechts) nach [KB07].

vermehrt Anwendung und werden zunehmend auch bei Produkten im Konsumbereich eingesetzt. Es handelt sich um spezielle berührungsempfindliche Oberflächen für die Eingabe von Daten mit Hilfe von Gesten. Größtenteils sind Bildschirm und Multi-Touch-Screen kombiniert und liegen übereinander. Somit ist der Benutzer in der Lage, dargestellte Elemente direkt auszuwählen oder zu manipulieren. Infolge einer bestimmten Elementreaktion wird zusätzlich der Eindruck des physischen Kontakts erzeugt. Spezielle kapazitive oder optische Touchscreens detektieren gleichzeitig mehrere Berührungspunkte, typischerweise die Fingerspitzen der Anwender. Im Gegensatz zu Systemen, die auf nur einen Eingabepunkt limitiert sind, können dadurch zusätzliche Bedienmethoden zum Einsatz kommen. Populäre Anwendungsbeispiele für Gesten sind das Skalieren oder Rotieren von visuellen Objekten, indem zwei Finger in bestimmter Weise zueinander bewegt werden. Neben der Eingabemöglichkeit durch Berührung einer touch-sensiblen Oberfläche mit mehreren Fingern oder anderen Elementen, können bei bestimmten Ausprägungen der Technik auch besondere Objekte³⁷ für die Eingabe genutzt werden. Dabei handelt es sich vorrangig um speziell geformte oder mit optischen Markern versehene Gegenstände, die festgelegte Aktionen oder digitale Komponenten repräsentieren. Die Ausmaße von Multi-Touch-Displays variieren stark und finden ebenso Einsatz bei kleinen tragbaren Geräten, wie bei großflächigen Wandinstallationen. Der Einsatzbereich derartiger Displays ist derzeit vor allem auf den Entertainmentbereich beschränkt. Dies ist unter anderem darauf zurückzuführen, dass der Hauptanteil eingesetzter Anwendungen weiterhin für traditionelle Eingabegeräte (Maus, Tastatur) entwickelt wird, derzeitige Multi-Touch-Geräte eine gebräuchliche Arbeitsstationen nicht ersetzen können und auf die vorherrschende Gewohnheit und bedingte Akzeptanz der Benutzer im fachmännischen bzw. industriellen Bereich. Multi-Touch selbst hat das Potenzial für NUI Anwendungen, jedoch beschränkt sich der momentane Einsatz meist auf die Interaktion mit herkömmlichen grafischen Benutzeroberflächen.

³⁷ engl. Tangible Objects

3 Stand der Technik

In diesem Kapitel wird ein zusammenfassender Überblick über den Stand der Visualisierung und Interaktion von Produktstrukturen vorgestellt. Der Schwerpunkt liegt hierbei auf den spezifischen Präsentationsformen innerhalb von erzeugenden CAD-Systemen und prozessbegleitenden PDM-Anwendungssystemen. Auf Forschungsansätze in den Bereichen der Visualisierung von Produktstrukturen und komplexen Datensystemen wird im Kontext dieser Arbeit nicht explizit eingegangen. An dieser Stelle sei daher auf relevante Literatur, wie [Les01, Shi10, Sch10, HTS09], verwiesen.

3.1 Produktstrukturen innerhalb von Anwendungssystemen

In der Produktentwicklung werden verstärkt erzeugende und prozessbegleitende informationstechnische Systeme zur Unterstützung der Arbeitsprozesse eingesetzt. Die erzeugenden Systeme orientieren sich dabei in der Regel am Kontext spezifischer Arbeitsphasen und sind auf diese abgestimmt. Hierzu zählen CAD-Systeme zur Erstellung von 3D-Volumenmodellen oder 2D-Zeichnungen, FEM-Systeme³⁸ zur Generierung von Berechnungsmodellen und DTP-³⁹ bzw. Office-Systeme für die Erzeugung von Arbeitsdaten. Die prozessbegleitenden Systeme beziehen sich hingegen auf den gesamten Produktentwicklungsprozess und sind nicht auf die einzelnen Arbeitsbereiche beschränkt. Zu diesen gehören vorrangig Anwendungen für das Produkt- und Produktdatenmanagement (PDM) und für die Produktionsplanung- und -steuerung (PPS). Im Kontext dieser Arbeit wird das Spektrum dieser Systeme auf geometriemodellierende CAD- und prozessbegleitende PDM-Systeme eingegrenzt. Weiterhin richtet sich der Fokus auf die Betrachtung der grafischen Präsentationsformen von Produktstrukturen.

3.1.1 CAD-Systeme

3D-CAD-Systeme zeichnen sich durch eine starke Durchdringung im Produktentwicklungsprozess aus und sind oft nicht auf die reine Geometrieerzeugung und -bearbeitung beschränkt. Die zum Einsatz kommenden Anwendungen integrieren weitere technologische Informationen, wie Designstudien, Konstruktionsabsichten und spezifische Modellberechnungen. Zur Verbindung dieser Informationen mit den zugehörigen Bauteilen und -gruppen werden mehrheitlich Produktstrukturen eingesetzt. Systemspezifische Konzepte und das hohe Funktionsspektrum führen jedoch zu einer heterogenen Abbildung der Produktstrukturmodelle. Diese werden folglich entsprechend des Anwendungskonzepts

³⁸ Finite Element Method (Methode der finiten Elemente)

³⁹ Desktop Publishing (Publizieren vom Schreibtisch aus)

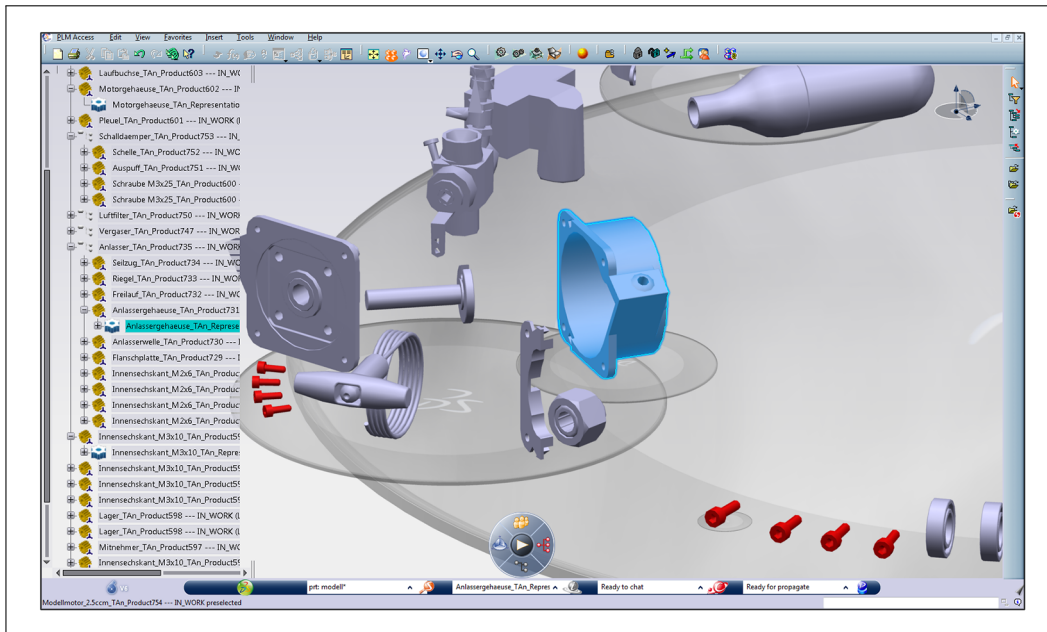


Abbildung 3.1: Exemplarische Visualisierung einer Produktstruktur (links) und des zugehörigen Geometriemodells (rechts) innerhalb eines CAD-Systems. Die räumliche Konfiguration der Geometriekomponenten orientiert sich an einem Strukturlayout des Produktstammbaums.

repräsentiert und präsentiert. Die häufigste Präsentationsform von Produktstrukturen ist die Abbildung der Aggregation mittels Indent-Layouts (Einschublayouts). Diese Layouts haben typischerweise die Form von vertikal ausgerichteten Baumdiagrammen, in denen die Bauteile und -gruppen anhand von Zeichenketten dargestellt werden. Die Strukturelemente des Baumdiagramms entsprechen den Elementen der Aggregation und sind horizontal bezüglich der Hierarchieebenen und Klassen eingerückt und geordnet. Abbildung 3.1 zeigt eine exemplarische Präsentation einer Produktstruktur in einem CAD-System.

Zur Steigerung der Wahrnehmung von Objekten werden Ikone, Farbkodierungen und Blinksignale eingesetzt. Diese fördern allgemein die Aufmerksamkeit beim Betrachter und unterstützen die Ermittlung und Identifikation von spezifischen Elementen. Die Ikone stellen außerdem einen Zusammenhang zwischen Produktstruktur und Geometrie her und bieten dem Produktentwickler mögliche Anhalts- und Bezugspunkte bei der Informationsanalyse. Bei den produktstrukturbezogenen Arbeitsprozessen kann der Benutzer Objekte gezielt auswählen und zugehörige Informationen und Metadaten mittels kontextsensitiver Dialogtechniken abfragen. Primär werden dabei nicht direkt visualisierte Informationen, wie bauteil- oder baugruppenspezifische Attribute, Assoziationen, Varianten und Änderungsstände verfügbar. Die Navigation mit der Strukturpräsentation folgt in CAD-Systemen überwiegend dem in Abschnitt 2.7 vorgestellten Konzept des Bildlaufs. Die Kombination von Indent-Layouts und Bildlauf ist ein verbreitetes Modell und wird

auch bei der klassischen Visualisierung und Interaktion mit Dateisystemen verwendet. Im Kontext der CAD-Systeme umfasst die Strukturnavigation hauptsächlich folgende Funktionen:

- Scrolling zum Verschieben des Anzeigebereichs auf Teilbereiche der Produktstruktur
- Skalierung zur Vergrößerung und Verkleinerung des Anzeigebereichs
- Positionierung des Anzeigebereichs
- Ein- und Ausblenden von Teilstrukturen

Die Präsentation und Interaktion einer Produktstruktur mittels Indent-Layout und Bildlauf unterliegt den in Tabelle 2.4 und in Abschnitt 2.7 dargelegten Eigenschaften. Die Ermittlung eines spezifischen Elements kann bei komplexen Strukturen eine große Anzahl an Interaktionen erfordern und stellt eine Belastung für den Benutzer dar.

Neben den verbreiteten Indent-Layouts existieren weitere Ansätze zur kombinierten Präsentation von Produktstruktur und Geometriemodell. Abbildung 3.1 zeigt eine Darstellung von Komponenten eines Geometriemodells, die anhand des Produktstammbaums geordnet sind. Die räumliche Konfiguration entspricht einem angepassten Balloon-Tree-Layout [MH98]. Bauteile und -gruppen können interaktiv entsprechend der Aggregation zu Gruppen zusammengefasst oder aufgeteilt werden. Die Präsentationsform ermöglicht dabei den direkten Bezug zwischen Geometriemodell und Produktstruktur. Die Konfiguration und Raumausnutzung des Layouts sind allerdings nur bedingt für komplexe Modelle geeignet. Die Anzeige aller atomarer Elemente führt zu Einschränkungen in der Übersichtlichkeit der Visualisierung und erfordert eine erhöhte Anzahl an Interaktionen bei der Informationsanalyse.

3.1.2 PDM-Systeme

PDM-Systeme sind Teile des betrieblichen Informations- und Koordinationssystems und dienen der konsistenten Speicherung und Verwaltung von Produktdaten in der Produktentwicklung. Die Daten werden transparent für die nachgelagerten Phasen des Produktlebenszyklus abgelegt und können über Vertriebswege bereitgestellt werden. Grundlage dafür ist ein integriertes Produktdatenmodell, das verschiedene Klassen, Objekte und Beziehungen beschreibt. Weitere Bestandteile des Produktdatenmodells sind Verweise auf Ressourcen und zusätzliche Informationen des Produktentwicklungsprozesses. PDM-Systeme bilden damit eine umfassende Integrationsplattform für die Verwaltung aller Produktdaten und die Basis zur Steuerung der Bearbeitung und Weiterleitung dieser Datenmenge. Zu den Funktionen einer PDM-Umgebung gehören neben Administrations-, Anpassungs- und Konfigurationswerkzeugen auch die Präsentation der elementaren Ausprägungen von Produktstrukturen. Des Weiteren werden verschiedene Sichten entlang des Produktentwicklungsprozesses unterstützt. Die Funktionalität eines

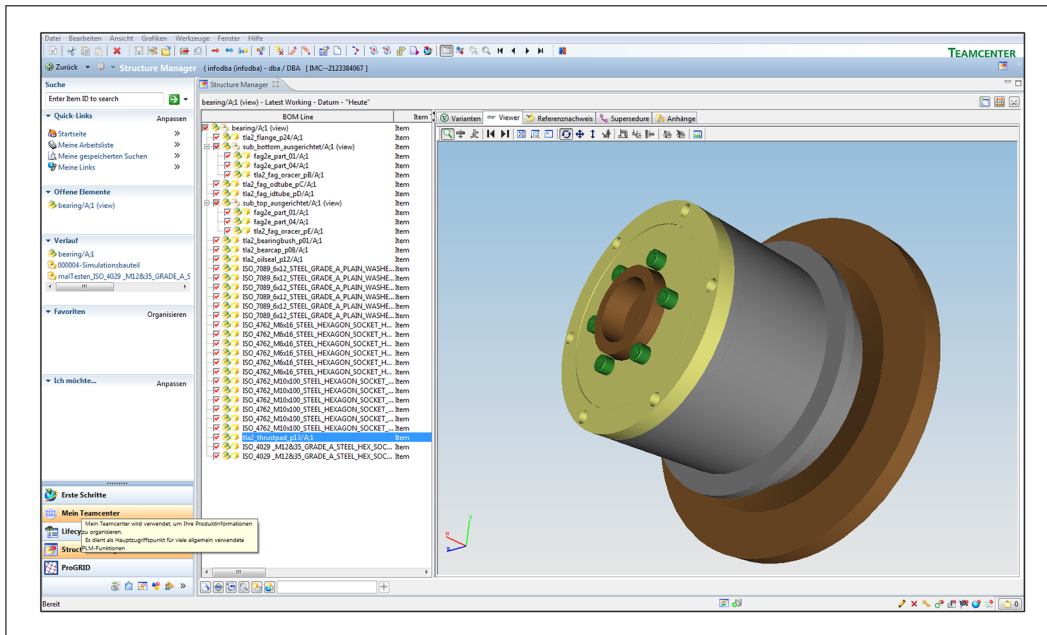


Abbildung 3.2: Exemplarische Visualisierung einer Produktstruktur (links) und des zugehörigen Geometriemodells (rechts) innerhalb eines PDM-Systems

übergreifend integrierenden PDM-Systems setzt sich folglich aus anwendungsbezogenen und anwendungsübergreifenden Funktionen zusammen [Les01].

Anwendungsbezogene Funktionen

- Produktstruktur- und Konfigurationsmanagement
- Klassifizierung und Teilefamilienmanagement
- Produktdaten- und Dokumentenmanagement
- Prozess- und Workflow-Management
- Projektdatenmanagement

Anwendungsübergreifende Funktionen

- Darstellung, Annotation
- Benutzermanagement
- Kommunikation
- Archivierung
- Datengewinnung
- Datenausgabe

Die grafische Präsentation von Produktstrukturen beschränkt sich in PDM-Systemen überwiegend auf die Visualisierung der Aggregation und Assoziation. Variation und Veränderung werden hingegen in der Regel in Form von Listen oder Tabellen dargestellt. Analog zu CAD-Systemen finden bei der grafischen Präsentation mehrheitlich Indent-

Layouts und die bereits beschriebenen Navigationsfunktionen Verwendung. Zusätzlich existieren integrierte oder gekoppelte Anwendungen zur Darstellung der Geometriemodelle. Diese ermöglichen entsprechend der spezifischen Anwendungsfunktionalität die gleichzeitige Visualisierung oder Hervorhebung der Geometriebestandteile bezüglich ausgewählter Elemente in der Produktstruktur. Die Präsentation einer Produktstruktur und des zugehörigen Geometriemodells ist in Abbildung 3.2 exemplarisch dargestellt.

Die systemspezifische Realisierung der PDM-Funktionalität führt zu einer Heterogenität der Produktdaten- und Produktstrukturpräsentation. Die Präsentationsformen der elementaren Ausprägungen von Produktstrukturen verschiedener PDM-Systeme können folglich voneinander abweichen. Die Trennung der Präsentation von Produktstruktur und Geometriemodell ist nur bedingt geeignet und stellt eine potenzielle Fehlerquelle dar. Der Benutzer muss zwischen den Präsentationsformen wechseln und sich fortwährend orientieren. Daraus resultierten zusätzliche Beanspruchungen während der Arbeitsprozesse.

Die enorme Datenmenge innerhalb der PDM-Systeme erfordert eine große Anzahl an Suchaktionen bei der Informationsanalyse. Folglich benötigen Konstrukteure ca. 30-40% der Arbeitszeit für die Recherche nach relevanten Komponenten [Kü04]. Derartige Suchanfragen sind typischerweise auf Texteingaben beschränkt. Der Benutzer muss eine Identifikationsnummer in eine Eingabemaske eingeben, und das Ergebnis wird anhand einer Liste ausgegeben.

4 Anforderungen und Handlungsbedarf

4.1 Eignungskriterien für Visualisierungssysteme

Das Ziel von Visualisierungsverfahren ist die Erkenntnisgewinnung und die Kenntnisgewinnung in Abhängigkeit von dem jeweiligen Einsatzfeld und der Zielgruppe [VDI03]. In diesem Zusammenhang beschreibt die **Erkenntnisgewinnung** die Umsetzung der Visualisierungsinhalte in subjektive Informationen beim Betrachter. Typische Nutzungsszenarien sind in diesem Fall die Verbesserung des allgemeinen Systemverständnisses, die Systemdiagnose zur Aufdeckung von Fehlern und die Unterstützung situationsbezogener Analysen durch Aufschlüsselung und Erklärung komplexer Systemzusammenhänge. Die **Kenntnisgewinnung** hingegen beschreibt den Prozess der Vermittlung von Informationen zwischen mindestens zwei Beteiligten. Dabei wird zum Beispiel die Schulung von Nutzergruppen unterstützt und führt zur verbesserten Kommunikation zwischen Experten, Novizen und anderen Bezugspersonen. Die Visualisierung ist dabei schwerpunktmäßig auf alle Nutzungs- und Anwendungsszenarien unter Berücksichtigung wahrnehmungspsychologischer Fähigkeiten, Wissen und Konventionen der Zielgruppe und weiteren in Abschnitt 2.4.1 vorgestellten Faktoren und Prinzipien abzustimmen. Grundsätzlich gilt, dass das verwendete Visualisierungssystem möglichst expressiv, effektiv und angemessen sein sollte, was wie folgt definiert wird:

Expressivität bezeichnet die unverfälschte Wiedergabe durch das Verfahren, wobei ausschließlich die in den Daten enthaltenen Informationen dargestellt werden.

Effektivität beschreibt die Zielstellung Informationen derartig zu vermitteln, dass diese möglichst intuitiv und ohne zusätzliche Erläuterungen vom Betrachter verstanden werden. Ein geeignetes Verfahren ist dabei vom Kontext und Anspruch der Visualisierung, von Wissen und Fähigkeiten des Nutzers und von der verwendeten Hardware und Software abhängig.

Angemessenheit bezieht sich auf Aufwand und Nutzen eines Visualisierungsverfahrens und ist bei der Erstellung und Interpretation in Relation zu setzen.

Der Erfolg oder Misserfolg des Einsatzes von Visualisierungsverfahren ist von weiteren Eignungsfaktoren abhängig, welche eine konkrete Auswahl begünstigen und Auswirkungen auf die allgemeinen Anforderungen haben.

Art und Struktur der Daten bestimmen die darin enthaltenen Informationen und sind in diesem Zusammenhang ausschlaggebend für den Grad der Expressivität. Es werden unter anderem Typ, Wertebereich, Dimension und die Gliederung der Daten unterschieden. Weiterhin werden die für eine Untersuchung wichtigen abhängigen und unabhängigen Variablen einbezogen.

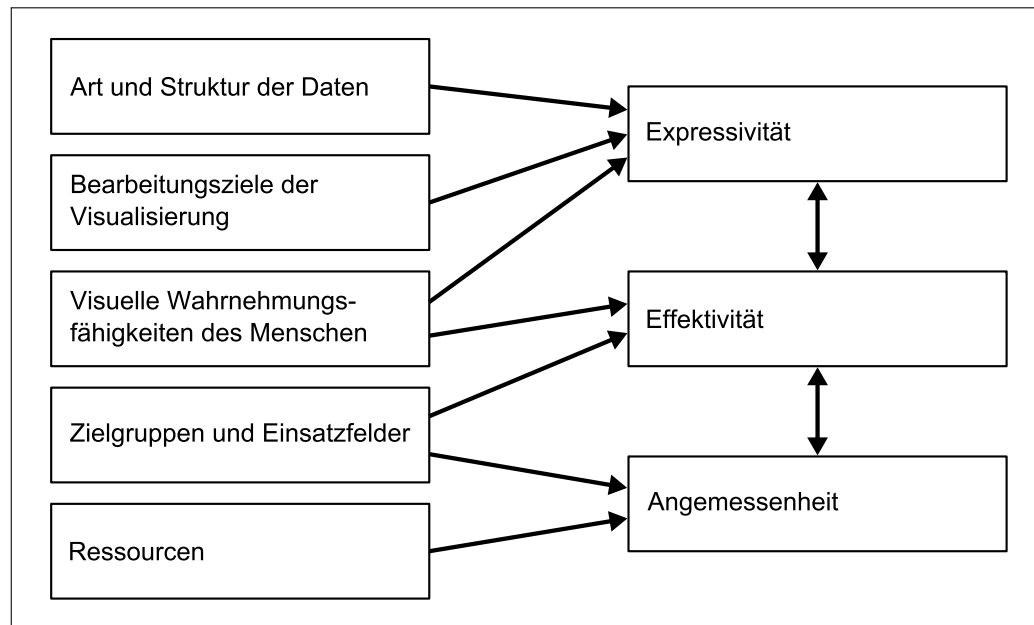


Abbildung 4.1: Wirkzusammenhänge zwischen Auswahlkriterien und allgemeinen Anforderungen an Visualisierungen nach [VDI03]

Bearbeitungsziele der Visualisierung werden häufig zunächst problembezogen (einsatzspezifisch) formuliert und dann als generalisierte visualisierungstechnische Ziele und Anforderungen, wie Identifikation, Korrelation, Vergleich, Häufigkeit, Gruppierung und Kategorisierung umgesetzt. Um die Expressivität einer Visualisierung bestimmen und beurteilen zu können, müssen diese Ziele explizit ausformuliert werden.

Visuelle Wahrnehmungsfähigkeiten des Menschen bilden, wie bereits angeführt, die Grundvoraussetzung für die Erzeugung und Präsentation von Visualisierungen und sind für die Effizienz der Umsetzung besonders zu beachten. Aspekte der menschlichen Kontrastempfindung, wie der Farb-, Textur-, Form-, Raum-, Bewegungs- und zeitliche Änderungswahrnehmung sind zu berücksichtigen. Bestimmte Reize können dabei allerdings zu Täuschungen und Illusionen führen. Dies liegt in der nicht objektiven, visuellen Wahrnehmung begründet und ist bei der Einhaltung des Expressivitätskriteriums zu beachten. Hierzu kommen weitere kognitive Faktoren, wie Aufmerksamkeit, Konzentration und Motivation der Benutzer. Idealerweise minimieren geeignete und effiziente Visualisierungsverfahren somit die kognitive Last für den Betrachter und erhöhen die Auswertegeschwindigkeit. Gleichzeitig werden Ermüdungserscheinungen und etwaige Fehlerquellen verringert.

Zielgruppen und Einsatzfelder können sehr unterschiedlich ausfallen und werden vor allem durch Kultur, Vorwissen, Vertrautheit mit bestimmten Darstellungsformen und die individuelle Kenntnis des Betrachtungsgegenstandes der beteiligten Personen geprägt. Der Abstraktionslevel der Visualisierung ist dem Fachwissen der Zielgruppe anzupassen, so dass bei geringen Vorkenntnissen eine geeignete, weniger abstrakte Darstellung auszuwählen ist.

Ressourcen werden bei jeder Visualisierung verbraucht und sind zum Beispiel Rechenzeit, Bildschirmfläche, Software und Planungs-, Gestaltungs-, Entwicklungs-, Modellierungs- und Testaufwand. Die Erfassung der Gesamtheit dieser Faktoren und die gegebenen Randbedingungen des Einsatzes, ermöglichen eine Kostenabschätzung für die Visualisierung und bilden die Grundlage für die Überprüfung der Angemessenheit.

Die angeführten Kriterien sind für qualitativ hochwertige Visualisierungen zu gewichten. Die Auswirkung der Kriterien und die Expressivität, Effektivität und Angemessenheit sind hierbei stets voneinander abhängig. Eine expressiv unangemessene Visualisierung kann daher nicht effektiv sein. Ferner haben ästhetische Aspekte bekanntlich große Auswirkung auf die Motivation des Betrachters und somit indirekten Einfluss auf die Effektivität der Visualisierung. Hinzu kommen gegebenenfalls Vorschriften des Corporate Design (Unternehmenserscheinung) oder kulturelle Unterschiede in der Zielgruppe, die zu beachten sind. Die Bildung konkreter, einfacher Regeln anhand dieser komplexen Fakten und Zusammenhänge ist bis heute im Allgemeinen nicht realisierbar und so ist deren Formalisierung nur ansatzweise möglich. Oft können nur Endnutzertests Aufschluss über die Auswahl und oder Kombination von Visualisierungsverfahren für ein konkretes Anwendungsproblem geben. Die im Rahmen dieser Arbeit entwickelten Konzepte werden daher im Kapitel 7 in kompakter Weise und anhand der hier erläuterten Kriterien diskutiert.

4.2 Integrationsaspekte

Aus Anwendersicht ist die Einbettung der Visualisierungsverfahren und -werkzeuge in die Unternehmenslandschaft von Produktion und Logistik ein wichtiger Aspekt für deren Akzeptanz und Einsetzbarkeit. Sowohl Daten und integrationsrelevante Werkzeuge der betroffenen Unternehmensbereiche, als auch deren Schnittstellen für den Datenaustausch und entsprechende Integrations- bzw. Kopplungskonzepte sind hierbei zu berücksichtigen. Die Visualisierung wird neben der Aufbereitung und Darstellung der Ergebnisdaten unter bestimmten Voraussetzungen auch für die Abbildung der unveränderten Eingangsdaten genutzt. Diese Rohdaten werden durch bestimmte Arbeitsabläufe und Werkzeuge generiert. In der Produktentwicklung gehören dazu verschiedene Werkzeugklassen, wie Planung, Design und Daten- und Dokumentmanagement, welche sich durch spezifische Alleinstellungsmerkmale und Datenformate unterscheiden.

CAD-Systeme werden in Unternehmen vorzugsweise für den rechnergestützten Entwurf und die Konstruktion von Produkt- und Anlagenlayouts eingesetzt. Durch entsprechende Zusatzmodule zählen dazu auch spezielle Anwendungsfelder wie die Installationsplanung. In diesem Kontext dienen diese Werkzeuge insbesondere als Quelle von Geometriedaten für eine realitätsnahe Visualisierung, eignen sich jedoch auch als Datenquelle für Strukturinformationen zu Produkten und Anlagen.

PDM-Systeme dienen als Basis für die Abbildung, Speicherung und Verwaltung von Daten und Dokumenten der Phasen eines Produktlebenszyklus. Dies umfasst vor allem produktdefinierende, -repräsentierende, -präsentierende Daten und Metadaten anhand eines integrierten Produktmodells und weiterführende Prozessmodelle in der Produktentwicklung. PDM-Systeme sind somit umfassende Produktdatenverwaltungsinstrumente, welche größtenteils auch CAD-System erzeugte Daten einbeziehen. In diesem Zusammenhang stellen diese Werkzeuge umfangreiche Produkt- und Strukturdaten und zusätzliche Leistungsdaten, Informationen und Metadaten zum Gesamtprodukt und dessen Komponenten bereit.

Im Unternehmen werden diese Werkzeuggruppen oftmals durch individuelle Softwarelösungen für spezielle Aufgaben ergänzt. Diese haben entweder die Form alleinstehender Neuentwicklungen oder basieren auf einer konfigurier- und erweiterbaren Arbeitsumgebung. Diese können teils selbst Visualisierungsaufgaben übernehmen oder sorgen mittels vereinheitlichter Schnittstellen idealerweise für bereits aufbereitete Eingangsdaten.

Schnittstellen zwischen den für die Visualisierung relevanten Werkzeugen sind maßgeblich in Anzahl und Art von dem Grad der Integration abhängig. Bei einer **vollständigen Integration** arbeiten alle Bestandteile auf einer gemeinsamen zentralen Datenbasis und sind in eine umfassende Anwendung eingegliedert. **Kopplung** hingegen beschreibt die datentechnische Verknüpfung von verschiedenen Werkzeugen, wobei die Daten werkzeugspezifisch dezentral gehalten werden. Hierbei verbindet die direkte Kopplung Systeme mit einem Verhältnis von 1:1. Bei speziellen, sequentiellen Prozessketten sind darüber hinaus Zusammenschlüsse bis zu einem n:m-Verhältnis möglich, bei denen die Daten zwischen allen beteiligten Systemen ausgetauscht werden können. Die Kopplung über eine Kommunikationsplattform (Middleware) erfordert weiterhin die vereinheitlichte Anbindung der betreffenden Systeme an eine gemeinsame Plattform. Diese übernimmt dann schließlich die kontrollierte Verteilung der Daten und regelt den Informationsaustausch zwischen den Systemen.

Bei einer Werkzeugkopplung ist der rein technische Austausch von Modelldaten nicht hinreichend. Ausgangsdaten eines Werkzeugs müssen daher oftmals einer **Datenaufbereitung** bezüglich Syntax und Semantik unterzogen werden, um das erneute Einlesen und Interpretieren eines anderen Systems zu ermöglichen. Für die Visualisierung sind hierbei generell die Zielsetzung, der geforderte Detaillierungsgrad und gegebenenfalls die verwendete Rechnerhardware und deren Performance zu berücksichtigen. Die **Datenkonvertierung** beschreibt dabei die Umwandlung, verlustbehaftet oder auch verlustfrei, der Ausgangsdaten eines Werkzeugs in ein Austauschformat (Export). Dieses Format kann folglich wiederum von einem anderen Werkzeug eingelesen werden (Import). Typischer Schwerpunkt ist dabei die Beachtung von Bezugsgrößen oder der Kontexte der Daten, welche bei der Konvertierung beibehalten werden müssen. Für hinreichende Leistungsfähigkeit und Visualisierungsqualität ist oft eine **Datenreduktion** erforderlich.

Diese umfasst sowohl die Reduktion von für die Visualisierung nicht relevanter Daten, als auch die Verkleinerung der Datenmenge. Beispielsweise werden hierbei nicht wahrnehmbarer Details, wie das Innenleben eines geschlossenen 3D-Geometriekörpers, eliminiert. Des Weiteren kann die Performance durch spezielle „Level-Of-Detail“-Verfahren (LoD) auch bei dynamisch wechselnden Anforderungen an die Visualisierung sichergestellt werden. Dabei werden unterschiedlich detaillierte Varianten einer Visualisierung erzeugt und in Abhängigkeit des Betrachtungspunktes abgerufen. **Strukturelle Datenänderungen** treten teilweise bei der Konvertierung von Daten in andere Präsentations- oder Repräsentationsformen auf. Beispielsweise werden in manchen Systemen bestimmte Darstellungsformen nur angenähert abgebildet, wodurch zur Gewährleistung der Datenqualität im Zielsystem eine Nacharbeit erforderlich wird.

Austauschformate dienen dem Transfer von Visualisierungs- und Produktdaten und lassen sich vor allem nach technischen Kriterien klassifizieren. Bei der Gestaltung unternehmensinterner Schnittstellen und deren Nutzung ist die eindeutige Definition, die regelmäßige Überwachung und die Entwicklung dieser verschiedenen Datenformate jedoch von hoher Bedeutung. In diesem Zusammenhang werden **standardisierte Formate** in nationalen oder internationalen Normen festgeschrieben. Je nach Anwendungsgebiet existieren hierfür verschiedene, werkzeugneutrale Formatspezifikationen die von Standardisierungsgremien definiert werden. Die von den Werkzeugen umgesetzte Qualität und Interpretation kann dennoch teils variieren, was unter Umständen auch bei standardisierten Schnittstellen zu Problemen führt. Bei **proprietären, offen gelegten Formaten** ist ein bestimmter Hersteller Eigentümer des Formates. In diesem Fall wird eine zusätzliche Dokumentation zur Verfügung gestellt, so dass andere Anbieter diese Formate ebenfalls unterstützen können. **Proprietäre, nicht offen gelegte Formate** sind hingegen „native“ Formate, welche die meisten Werkzeuge nutzen, um anwendungsspezifische Daten zu speichern. Dabei sind lediglich diese Werkzeuge bzw. Werkzeuge der gleichen Gruppe in der Lage diese Formate zu lesen und zu schreiben. Unabhängig davon existieren weitere De-facto-Standards, deren Bedeutungen in erster Linie durch den kommerziellen Erfolg eines Produkts bestimmt werden. Diese entstehen somit als proprietäres Format und können gegebenenfalls den Status eines standardisierten Formats erreichen.

4.3 Erstellungsaufwand und Nutzen

Die Visualisierung ist als interdisziplinäres Kommunikationswerkzeug zu verstehen und wird zumeist nicht zum Selbstzweck eingesetzt. Das eigentliche Maß für den Nutzen einer Visualisierung ergibt sich daher aus den in Abschnitt 4.1 erläuterten Zielsetzungen des Erkenntnisgewinns und der Kenntnisvermittlung. Einflussfaktoren für eine nutzbringende Vermittlung von Informationen sind dabei insbesondere der Erstellungs- und der Interpretationsaufwand der Zielgruppe. Für den Erstellungsaufwand sind verschiedene Aspekte zu berücksichtigen, wobei ein Hauptschwerpunkt die Integration des Visualisie-

ungsverfahrens ist. Die Produktentwicklung erfordert häufig die Arbeit mit verschiedenen Anwendungen und Systemen, die teils sogar parallel zu bedienen sind. Dies erfordert hohen Organisationsaufwand und kognitive Last beim Benutzer. Die Visualisierung ist daher im Allgemeinen möglichst eng mit bestehenden Systemen zu verbinden, um Nutzern die Informationen in gewohnter Arbeitsumgebung zu vermitteln und die Akzeptanz für ein zusätzliches Werkzeug zu steigern. Die Auswahl und Bereitstellung von Eingangsdaten für die Visualisierung erfordert unterschiedliche Schritte. Hierbei finden sowohl rein manuelle, aber auch bis zu halb- oder vollautomatische Verfahren Anwendung, wobei die Durchführbarkeit und der dabei entstehende Zeitaufwand stark von dem Wissenstand des Nutzers abhängt. Neben gegebenenfalls zusätzlich nötiger Datenaufbereitung oder -umstrukturierung, haben Systemreaktionszeiten hierbei ebenso gewichtige Auswirkungen. Inakzeptable Ausführ- und Ablaufgeschwindigkeiten können daher die Anwendbarkeit und Zweckmäßigkeit für den Benutzer minimieren. Im Gegensatz dazu kann in manchen Anwendungsfällen der Nutzen eines Visualisierungsverfahrens steigen, wenn die Darstellung und Informationen auch anhand gewonnener Kenntnisse manipulierbar, verifizierbar und weiter verwertbar sind. Je nach Anbindung mit dem Ursprungssystem der Daten sollten vorgenommene Veränderungen und Festlegungen übertragbar sein, um diese nicht nur auf die Visualisierung zu beschränken.

4.4 Handlungsbedarf

Eine zentrale Herausforderung im Bereich der virtuellen Produktentwicklung ist das Management produktbezogener Daten. Diese Problematik gewinnt insbesondere im Zusammenhang mit der fortwährend ansteigenden Produktkomplexität und der zunehmenden Visualisierung des Produktentwicklungsprozesses an Bedeutung. Zur Gewährleistung der Qualität in der Produktentwicklung ist es daher notwendig, die Produktentwickler mit geeigneten Werkzeugen zur Visualisierung von Produktdaten zu unterstützen. Der Anspruch an derartige Werkzeuge folgt Kriterien wie der Integration in bestehende Systeme, der transparenten Abbildung von Entwicklungsständen und die Hervorhebung von etwaigen Entwurfs- und Planungsfehlern.

In [DI10] wurde mittels einer qualitativen Inhaltsanalyse ermittelt, dass die Möglichkeiten der Informationsdarstellung und Interaktion in bestehenden PDM-Systemen unzureichend sind. Innerhalb einer Studie wurde dabei die Funktionalität bestehender Systeme in Bezug auf innere Informationsverarbeitung und Denkvorgänge von den Teilnehmern überwiegend negativ eingeschätzt. Die für die vorliegende Arbeit wesentlichen Ergebnisse der Studie sind in den Tabellen 4.1 und 4.2 zusammengefasst. Es werden die positiven und negativen Aussagen in Bezug auf Darstellung, Verarbeitung und Interaktion von Informationen in bestehenden PDM-Systemen vorgestellt.

In der Studie ist weiterhin die Überladung der grafischen Benutzungsschnittstelle der PDM-Systeme als Schwachstellen identifiziert worden. Entsprechend den Beobachtun-

	Informationsdarstellung	Informationsverarbeitung	Informationsinteraktion
Positive Aussagen	Man kann sowohl die Zusammenhänge der Bauteile innerhalb der hierarchischen Baumstruktur erkennen, als auch die Geometrie einer beliebigen Bauteilgruppe im PDM-System sehen.	Man kann Dinge gut merken und einordnen, wenn man ein Bild von ihnen hat.	Mit einem Doppelklick in der Baumstruktur wird das Vorschaufenster mit der 3D-Darstellung geöffnet.
Anteil der Aussagen	56%	100%	30%

Tabelle 4.1: Aussagen mit positiver Valenz zu bestehenden PDM-Systemen nach [DI10]

	Informationsdarstellung	Informationsverarbeitung	Informationsinteraktion
Negative Aussagen	Das Interface ist überladen, unübersichtlich und beinhaltet sehr viele tiefe Verschachtelungen. Individuelle Ansichten sind nicht möglich.	Die eigene Sichtweise des Problems lässt sich in der gegebenen Darstellung nicht wiederfinden. Dadurch entstehen Unsicherheiten, Fehlentscheidungen und mangelnde Motivation.	Die Oberfläche bietet zu viele Möglichkeiten zum Klicken an, teilweise sind diese Schaltflächen nicht funktional gekennzeichnet. Man kann ausschließlich aufklappen und scrollen. Es gibt keine Eingabehilfen bei Textfeldern.
Anteil der Aussagen	43%	90%	62%

Tabelle 4.2: Aussagen mit negativer Valenz zu bestehenden PDM-Systemen nach [DI10]

gen wurde folglich eine Reduktion der Funktionalität auf die tatsächlich relevanten Funktionen, wie Datenstruktur anlegen, Daten einfügen und Datenermittlung und -bearbeitung empfohlen. Darüber hinaus wurde die generelle Präsentationsform mittels eines vertikal gestreckten Strukturlayouts (Indent-Layout) und die Interaktionsmethodik auf dem Prinzip des Bildlaufs kritisiert. Neben der unverhältnismäßigen Anzahl von Scrolling-Aktionen und Texteingaben empfanden die Probanden den Anteil der gleichzeitig dargestellten Informationen als zu gering. In diesem Zusammenhang wurden die daraus resultierenden Beeinträchtigungen in Denkvorgängen und im Arbeitsablauf explizit angesprochen.

Die vorliegende Arbeit orientiert sich an den generellen Optmierungsaspekten in [DI10], folgt aber einer anderen Herangehensweise. Für die Reduzierung der kognitiven Belastung und zur Förderung von Effektivität und Effizienz in den Arbeitsprozessen, werden verschiedenartige Visualisierungskonzepte und Interaktionsmethoden untersucht und prototypisch umgesetzt. Mit der Einbettung in ein flexibles System werden diese Konzepte nach spezifischen Anforderungen und Eignungskriterien bemessen und die Potenziale in Verbindung mit innovativen Präsentationsmedien und natürlichen Benutzungsschnittstellen diskutiert. **Handlungsbedarf** besteht dabei primär in der Realisierung dieses Vorgehens und der zugehörigen Bestandteile, welche durch folgende Punkte gekennzeichnet sind:

- Identifikation und Analyse geeigneter Visualisierungs- und Interaktionstechniken
- Berücksichtigung der elementaren Ausprägungen und weiterer spezifischer Anforderungen von Produktstrukturen
- Kombination von Geometriemodellen und Produktstrukturen
- Komplexitätsbeherrschung und Skalierbarkeit
- Allgemeine Gebrauchstauglichkeit und Erfüllung der Eignungskriterien
- Intuitive Bedienung und Orientierung nach kollaborativen Arbeitsprozessen
- Berücksichtigung eines angemessenen Zusammenhangs zwischen Erstellungsaufwand und resultierendem Nutzen
- Reduktion der Arbeitsbelastung
- Adaption an Zielgruppe und Einsatzfelder
- Modularität, Erweiterbarkeit und Systemunabhängigkeit
- Integration in Produktentwicklungsprozesse

5 Entscheidungsfindung und Konzeption

5.1 Entscheidungsfindung

Der Schwerpunkt dieser Arbeit ist die Auswahl und Analyse von geeigneten Visualisierungs- und Interaktionsmethoden für Produktstrukturen. Insbesondere werden hierbei Konzepte in Bezug auf Abbildung und Umgang mit dem hierarchischen Ordnungssystem von Produkten, dem Stammbaum bzw. der Aufbauübersicht, bestehend aus Bauteilen und Baugruppen in Betracht bezogen. Die allgemeingültige Methodik der Visualisierungspipeline bildet hierbei die Basis für die Konzeption und Umsetzung von Prototypen, anhand derer verschiedene Techniken evaluiert werden. Die in Kapitel 2 dargelegten theoretischen Grundlagen, Richtlinien und Referenzen dienen in diesem Zusammenhang der Orientierung für Entwurf und Umsetzung. Zentrales Bestreben ist die Entwicklung eines geeigneten Modells, welches den Produktentwickler durch intuitive Bedienung und Verbindung von Produktstruktur und Geometriemodellen bei seinen Arbeitsprozessen unterstützt. Die Erreichung der Zielstellung und die Eignung und Angemessenheit der entwickelten Konzepte werden in einer anschließenden Validierung diskutiert.

5.1.1 Auswahl der Visualisierungskonzepte

Im Abschnitt 2.6 wurde bereits eine allgemeine Klassifikation für Visualisierungstechniken beschrieben und weitere Unterscheidungsmerkmale, wie die Dimensionalität angeführt. Aufgrund der verschiedenartigen Kategorien und facettenreichen Ausprägungen ist die direkte Identifikation von besonders geeigneten Konzepten für die Abbildung von Produktstrukturen allerdings nur bedingt möglich. Folglich wird die Diskussion und Ermittlung im Rahmen dieser Arbeit entsprechend der Aggregation von Bauteilen und -gruppen auf zweckmäßige Techniken für hierarchische Datenstrukturen eingegrenzt. Die Diagrammklassen, Kapselung und Knoten-Kanten dienen in diesem Zusammenhang als vordefinierte Auswahl und werden anhand jeweiliger Ausprägungen auf deren Eignung untersucht. Durch experimentelle Studien und dem Stand der Technik, wird hierbei für jede Klasse eine repräsentative Layouttechnik ausgewählt und an die Anforderungen der Produktstrukturen angepasst.

Zur Gewährleistung einer möglichst generellen Herangehensweise werden für die Auswahl der Visualisierungskonzepte ausschließlich **2D-Layout-Algorithmen** berücksichtigt. 2D-Layouts können typischerweise mittels Erweiterungen und gestalterischer und technischer Modifikationen in 3D-Layouts überführt werden. 3D-Visualisierungstechniken sind hingegen durch die Nutzung des zusätzlichen Freiheitsgrades nur bedingt ohne Einschränkungen, in den 2D-Raum zu konvertieren. Ausgehend von dieser Charakteristik werden im Rahmen der Arbeit folglich 2D-Konzepte als abstrakte Basis für weitere

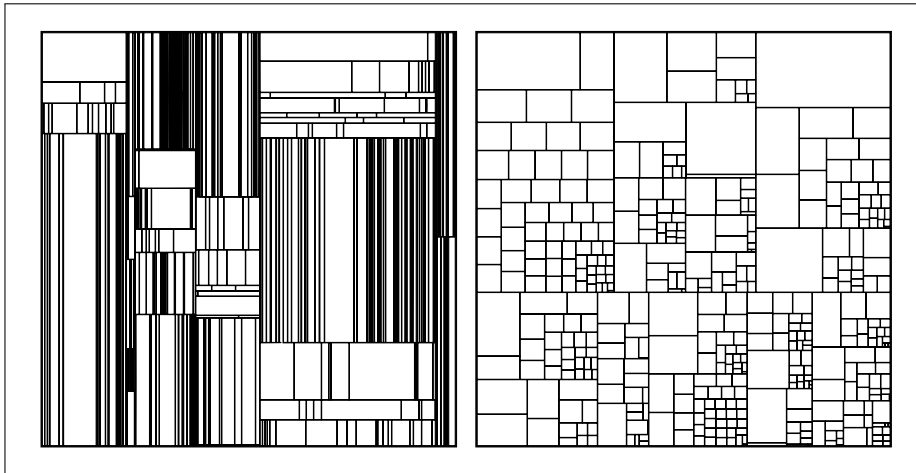


Abbildung 5.1: Interpretationsprobleme bei Slice-And-Dice und Squarified Treemap-Layouts nach [BD05]

Verbesserungen und Funktionssteigerungen untersucht. Die gewonnenen Erkenntnisse bezüglich der 2D-Konzepte sind nicht auf diese limitiert und allgemein auf entsprechende 3D-Techniken übertragbar. Grundlegende Richtlinien und Optimierungen in Bezug auf Verständlichkeit und Komplexitätsbeherrschung können so unter bestimmten Voraussetzungen direkt übernommen und mit der Dimensionalitätssteigerung weiter spezialisiert werden. Beispielsweise könnten mit dem Übergang von 2D zu 3D weitere Aspekte, Datenwerte und Beziehungen integrierbar werden.

Zur Generierung eines **Kapselung-Layouts** (Treemaps) ist es notwendig, eine Unterteilungsmethodik zu definieren. Ein klassisches Konzept für die Erzeugung von rechteckigen Unterteilungen ist beispielsweise der Slice-And-Dice-Algorithmus [Shn91]. Die generierten Rechtecke besitzen allerdings ein sehr einseitiges Seitenverhältnis, wodurch die Interpretation des Layouts beeinträchtigt wird. Andere Techniken, wie die Squarified-Treemaps [BHW00] erzeugen hingegen Rechtecke mit ausgeglichenem Seitenverhältnis. Der Nachteil dieser Treemaps ist jedoch die hohe Ähnlichkeit von Teilbereichen des Layouts, wodurch wiederum die Differenzierung erschwert wird. In Abbildung 5.1 wird die Problematik dieser Konzepte exemplarisch dargestellt. In diesem Zusammenhang werden in [BD05] generelle Richtlinien und Optimierungskriterien für die Generierung von Treemap-Layouts vorgestellt. Die Kriterien beziehen sich insbesondere auf die Form der Teilregionen des Layouts und werden wie folgt beschrieben:

1. Die rekursive Unterteilung muss den zur Verfügung stehenden Layoutbereich vollständig ausnutzen, ohne dabei Löcher oder Überlappungen zu erzeugen.
2. Alle Teilregionen sollten ein möglichst ausgeglichenes Seitenverhältnis besitzen.
3. Zur Erzeugung einer eindeutigen Strukturabbildung sollten Elemente einer Klasse nicht zusammengefasst werden.
4. Zur Förderung der Differenzierung von Teilregionen sollten diese distinkte Formen besitzen.

Die Erfüllung dieser Kriterien wird in [BD05] durch die Adaption der Voronoi-Tessellierung zur Generierung von **Voronoi-Treemaps** erreicht. Die Forschungsarbeit beschreibt die iterative und rekursive Berechnung der Voronoi-Tessellierung zur Erzeugung von Unterteilungen mit polygonalen Flächen. Die resultierenden Voronoi-Treemaps sind eine vollständige Abbildung der Ausgangsstruktur mit effizienter Raumausnutzung. Des Weiteren haben die Teilregionen ein annähernd gleichgewichtiges Seitenverhältnis und eine leicht differenzierbare Form, wodurch die angeführten Kriterien erfüllt werden. Wie aus Tabelle 5.1 ersichtlich, können mittels Voronoi-Treemaps vergleichsweise gut geeignete dynamische Strukturabbildungen erzeugt werden. In der Folge werden Voronoi-Treemaps als repräsentatives Konzept für die Kategorie Kapselung ausgewählt. Die im Rahmen der Arbeit verfolgte prototypische Umsetzung und spezifische Adaption wird in Abschnitt 6.2 beschrieben.

	Durchschnittliches Seitenverhältnis	Stabile Aktualisierungen	Stabile Skalierung
Slice&Dice [Shn91]	100+	✓	×
Squarified [BHW00]	1,3	×	✓
Ordered [BSW02]	2,8	×	×
Strip [BSW02]	2,5	×	×
Spiral [TS07]	2,5	×	×
Voronoi [BD05]	1,3	✓	✓

Tabelle 5.1: Eigenschaften von Treemap-Algorithmen nach [SFL10]

Die Auswahl des **Knoten-Kanten-Konzepts** folgt einer explorativen Herangehensweise. In experimentellen Umsetzungen werden zunächst verschiedenartige Knoten-Kanten-Layouts generiert und unter diversen strukturellen Voraussetzungen analysiert. Eine Grundbedingung ist hierbei die Vermeidung von Überlappungen und Überschneidungen der Knoten und Kanten. Gleichzeitig bildet die Symmetrie und ästhetische Wirkung der räumlichen Konfiguration der Layoutkomponenten ein wichtiges Auswahlkriterium. Zur Gewährleistung der Flexibilität und Skalierbarkeit der Algorithmen werden darüber hinaus die Auswirkungen von verschiedenartigen Ausgangsstrukturen untersucht. Extreme Ausprägungen der Strukturen, wie beispielsweise mit sehr geringer oder großer Strukturtiefe, dienen als Basis für die Layoutgenerierung und die resultierenden Abbildungen werden bezüglich der Interpretierbarkeit bemessen. Ein simpler und effektiver Algorithmus für die Erzeugung von Knoten-Kanten-Layouts ist der Layering-Algorithmus [TDBET98]. Bei diesem Algorithmus werden alle Elemente einer Hierarchieebene typischerweise einer vertikalen Stufe zugeordnet und mit einem berechneten Abstand horizontal verteilt. Die vertikalen Stufen werden hierbei mit zunehmender Distanz vom Wurzelement organisiert bis die gesamte Struktur abgebildet ist. Die resultierenden Layouts des

Layering-Algorithmen können unter den genannten extremen strukturellen Voraussetzungen allerdings sehr einseitige Proportionen aufweisen. In Übersichtsdarstellungen gleichen die Strukturabbildungen vertikalen bzw. horizontalen Linien, wodurch die Interpretation erheblich beeinträchtigt wird.

Ein weiterer Ansatz ist der Radial-Drawing-Algorithmus [TDBET98]. Bei diesem Algorithmus bildet das Wurzelement den Ursprung des Layouts, um den alle Elemente in konzentrischen Kreisen und mit steigendem Radius angeordnet werden. Die Elemente einer Hierarchieebene werden hierbei jeweils einem Kreis zugeordnet und auf der Kreislinie verteilt. Wie in Abbildung 5.2 illustriert, können auch bei diesen Layouts unter bestimmten Voraussetzungen unausgewogene Strukturabbildungen entstehen. Die experimentelle Umsetzung des Algorithmus tendiert dazu, die Strukturelemente nicht entsprechend der Strukturklassen anzuordnen und erschwert somit unter anderem die Analyse des Elementkontextes.

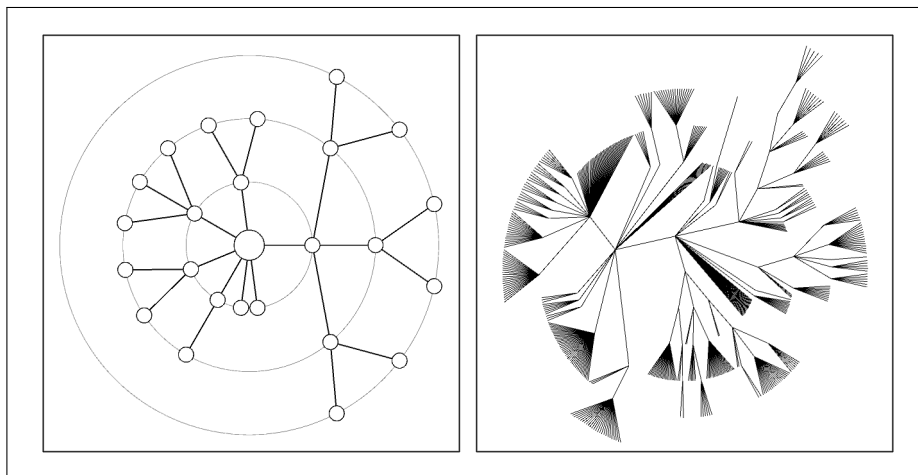


Abbildung 5.2: Erzeugung eines Knoten-Kanten-Layouts mittels Radial-Drawing-Algorithmus. Abgebildet sind eine simple Strukturabbildung mit Hervorhebung der Hierarchieebenen (links) und ein unausgeglichenes Layout mit komplexer Ausgangsstruktur (rechts).

Ein andersartiges Verfahren zur Generierung von Knoten-Kanten-Layouts ist die Verwendung eines physikalischen Systems. In diesem Zusammenhang nutzen **Force-Directed-Algorithmen** ein der Struktur entsprechendes Kräftesystem und berechnen in einem iterativen Prozess die optimale räumliche Konfiguration der Strukturelemente. Das Resultat sind in der Regel Layouts mit hoher Symmetrie und eine strukturbezogene Gruppierung der Layoutkomponenten. Die Möglichkeit zur Parametrisierung der Algorithmen unterstützt die Skalierbarkeit und Flexibilität der Layoutgenerierung. Für die Kategorie Knoten-Kanten werden folglich die Force-Directed-Layouts als repräsentatives Konzept ausgewählt und im Kontext der Arbeit angepasst. Die prototypische Umsetzung dieses Konzepts wird in Abschnitt 6.2 im Rahmen der systemtechnischen Realisierung fortführend beschrieben.

5.2 Konzeption des Visualisierungs- und Interaktionssystems (VIS)

Das übergeordnete Ziel dieser Arbeit ist die Identifikation, prototypische Umsetzung und Diskussion von potenziellen Optimierungsmaßnahmen zur Visualisierung und Interaktion von Produktstrukturen. Das Konzept eines **Visualisierungs- und Interaktionssystem (VIS)** dient hierbei zur Erfüllung der gesetzten Zielstellung. Die Entwicklung des VIS orientiert sich an den vorgestellten Grundlagen, Anforderungen und Entscheidungen der vorangegangenen Kapitel und umfasst alle zu untersuchenden Bestandteile für die Darstellung und den Umgang mit den Produktstrukturen. Hauptelemente sind die in Abschnitt 5.1.1 vorgestellten Visualisierungskonzepte und die zugehörige Interaktionsmethodik zur Unterstützung einer dynamischen Informationsanalyse. Eine modulare Systemarchitektur und flexible Wirkbeziehungen ermöglichen die Validierung der einzelnen Komponenten und individuellen Anpassungen. Die wesentlichen Elemente der Konzeption des VIS werden in diesem Kapitel zusammengefasst.

Das **Interaction Design** (IxD, dt. Interaktionsgestaltung) beschreibt die Gestaltung der Funktionalität, des Verhaltens, der Interaktionsprozesse und der endgültigen Ausgestaltung von Systemen. Der Bereich des Interaction Design umfasst folglich alle die Wahrnehmung und das Erlebnis der Benutzer betreffenden Bestandteile eines Systems. Die Konzeption und Entwicklung der Funktionen und Komponenten des VIS richten sich vorrangig nach den allgemeinen Richtlinien zur Gestaltung benutzerorientierter, interaktiver Systeme der Standardreihe EN ISO 9241. Insbesondere in [DIN98] und [DIN08] werden allgemeine ergonomische Anforderungen und Grundsätze für die Gestaltung von Software und Anwendungssystemen beschrieben, die unabhängig von einer systemspezifischen Dialogtechnik sind. Es werden darin unter anderem Kriterien der Anwenderführung, der Menü- und Dialoggestaltung und der Farbdarstellung behandelt. Die mit dem Interaction Design im Kontext der Arbeit verfolgten Ziele, richten sich neben den genannten Richtlinien und in Abschnitt 4 beschriebenen Anforderungen auch primär nach den Bedingungen des Eignungsvergleichs der Visualisierungskonzepte. In diesem Zusammenhang werden die Interaktionsfunktionalität, Dialogverfahren und grafische Bedienelemente auf relevante Komponenten reduziert, um eine simple und effiziente Handhabung mit hohem Bedienkomfort zu realisieren. Insbesondere die Interaktionsmethoden in Bezug auf die Visualisierungskonzepte werden hierbei für eine dynamische Informationsanalyse konzipiert und entsprechend gestaltet. Der Anwender soll dabei die gewünschten Aufgaben effektiv lösen und besser mit der Software arbeiten können. Das Konzept der **Gebrauchstauglichkeit** eines Systems fasst die dafür notwendigen Faktoren, Kriterien und deren Beziehungen allgemeingültig zusammen und dient als Richtlinie für die Entwicklung der Bestandteile des VIS. Das generelle Modell

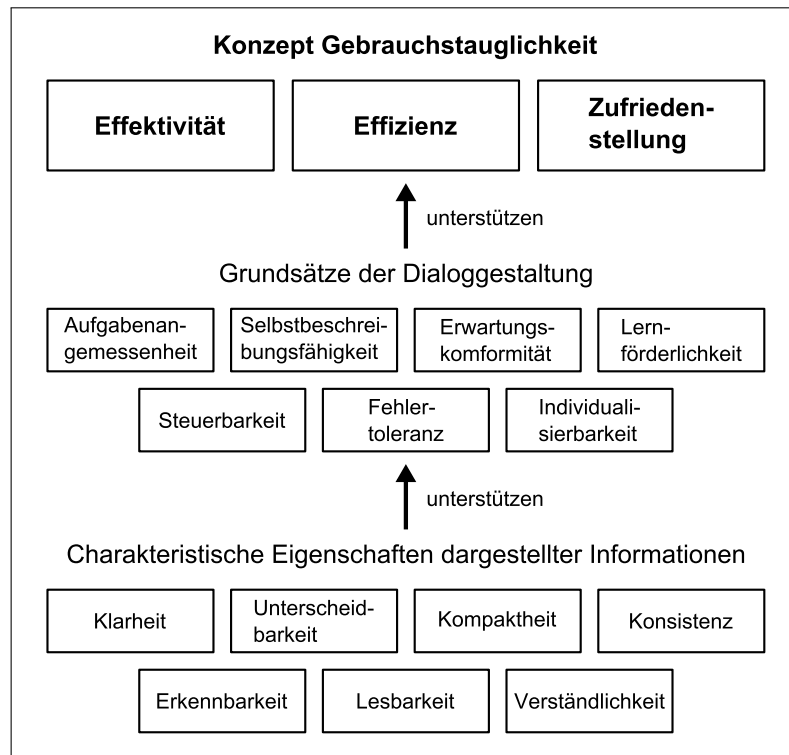


Abbildung 5.3: Beziehungen von Faktoren und Kriterien bezüglich der Gebrauchstauglichkeit eines Systems nach [DIN08]

der Gebrauchstauglichkeit ist in Abbildung 5.3 illustriert und kann nach [DIN08] wie folgt beschrieben werden:

„Das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen.“

Die Kombination und das Zusammenwirken der Komponenten des VIS erfordern eine flexible und modulare **Systemarchitektur**. Grundvoraussetzung hierfür ist die Unterteilung des Systems in funktionale Bestandteile und die Abstraktion von system- oder anwendungsspezifischen Elementen. Die Einteilung des VIS umfasst in diesem Zusammenhang die grundlegenden Module wie Integrations- bzw. Datenextraktionsschnittstelle, Visualisierungsfunktionalität, Datenhaltung und Benutzungsschnittstelle. Diese Komponenten werden in einem Basissystem integriert und zusammengefasst. Das Basissystem stellt weitere Werkzeuge zur Verfügung und ermöglicht die einheitliche Kommunikation und den Datenaustausch zwischen den Komponenten. Zur Gewährleistung der übergeordneten Anforderungen in Bezug auf Modifizierbarkeit, Erweiterbarkeit und Wiederverwendbarkeit des VIS und der zugehörigen Bestandteilen, wird im Kontext der Arbeit weiterhin das **Model-View-Controller (MVC)** Architekturmuster angewendet. Das MVC-Konzept strukturiert eine Softwareentwicklung in die drei Einheiten Datenmodell (Model), Präsentation (View) und Programmsteuerung (Controller). Das Modell ist hierbei unabhängig von der Präsentation und Steuerung und umfasst die darzustellenden Daten. Eine Änderung von

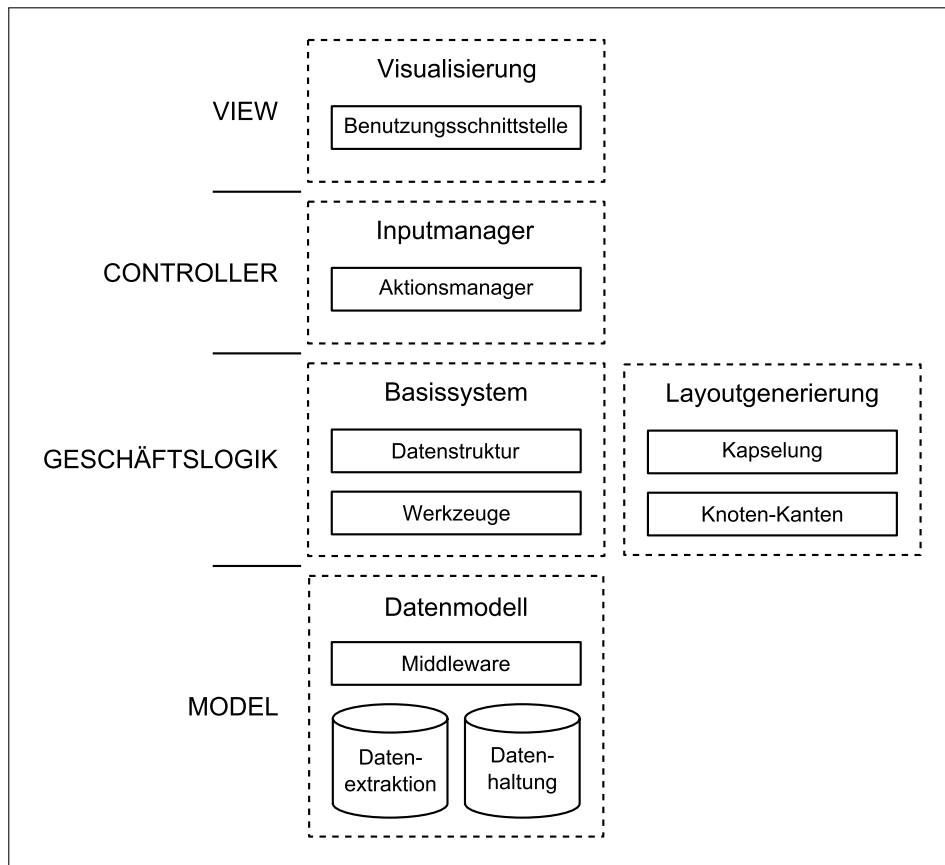


Abbildung 5.4: Schematische Darstellung der Systemarchitektur des VIS und Einteilung in funktionale Elemente des MVC-Konzepts

relevanten Daten im Modell wird durch spezielle Mechanismen an die entsprechenden Einheiten weitergegeben und verarbeitet. Die Präsentationsschicht ist für die Darstellung der benötigten Daten aus einem zugehörigen Modell verantwortlich und ruft diese nach Veranlassung ab. Zusätzlich nimmt die Präsentation Benutzerinteraktionen entgegen und übergibt diese der Steuerung zur Weiterverarbeitung. Die Steuerung verwaltet in diesem Zusammenhang eine oder mehrere Präsentationen und bestimmt anhand von gefilterten Benutzeraktionen, welche Daten im Modell zu ändern sind. Die eigentliche Manipulation der Daten wird hierbei der Geschäftslogik der jeweiligen Anwendung überlassen und wird typischerweise vom MVC-Konzept getrennt. Das verfolgte Ziel ist ein flexibler Programmentwurf, der beispielsweise die Entwicklung verschiedenartiger Oberflächen für ein System und die Adaption an bestimmte Systemvoraussetzungen unterstützt. Im Kontext des VIS ermöglicht das MVC-Entwurfsmuster unter anderem die gleichzeitige Integration verschiedener Visualisierungskonzepte auf Grundlage einer einheitlichen Datenbasis und die Abgrenzung zur Datenextraktion und -haltung. Das hierbei verfolgte Konzeptionsschema ist in Abbildung 5.4 illustriert.

Die technischen Rahmenbedingungen der Entwicklung des VIS erlauben den Einbezug von **Multi-Touch Technologie** für die Konzeption von alternativen und divergenten Interaktionsmethoden. Hierbei werden sowohl simple Aktionsformen, als auch komplexe

Interaktionsparadigmen mit einer spezifischen Semantik in Betracht gezogen. Die Kombination der einzelnen Interaktionsschemen ermöglicht die Einbindung weiterer Potenziale in Bezug auf den intuitiven und natürlichen Umgang mit dem VIS. Im Rahmen der Arbeit umfasst die Konzeption der Multi-Touch-Interaktion primär die Abbildung des Aktionspektrums auf gebräuchliche Gestenmuster im Umgang mit Multi-Touch-Geräten. Dies fördert die Anwenderakzeptanz von Benutzern mit Vorwissen im Multi-Touch-Bereich und minimiert gleichzeitig Einarbeitungszeit durch den Einsatz erprobter Konzepte. Zu den klassischen Gestenmustern zählen beispielsweise spezielle Eingabemethoden und Aktionsabfolgen für die Skalierung, Rotation und Auswahl von Anwendungskomponenten. Diese Methoden haben sich in Folge der steigenden Vermarktung von Multi-Touch-Geräten durchgesetzt und können anwendungs- und systemunabhängig betrachtet werden. Neben der generellen Erweiterung der Bedienelemente des VIS besteht der Schwerpunkt in der Adaption dieser Muster auf die spezifische Interaktionsfunktionalität mit den Visualisierungskonzepten. Die zweckmäßige Anpassung der grundlegenden Interaktionskonzeption ermöglicht daraufhin Individualisierungen und die Ergänzung weiterer Gesten.

6 Systemtechnische Realisierung

In diesem Kapitel wird die systemtechnische Realisierung des **Visualisierungs- und Interaktionssystems (VIS)**, dessen Hauptkomponenten und darauf aufbauende informationstechnische Konzepte vorgestellt. Die Entscheidungsfindung zu eingesetzten Werkzeugen und Technologien orientiert sich hierbei vorrangig an offenen Standards, Open Source⁴⁰ und freien Formaten. Dies begünstigt, neben einer modularen Systemarchitektur nach dem MVC-Prinzip, die Flexibilität, Wiederverwendbarkeit und Erweiterbarkeit der entstandenen Softwarekomponenten und fördert zudem die Portabilität und den Einsatz in multiplen Anwendungsfeldern.

Anhand der im Kapitel 5 getroffenen Entscheidungen wird das **Basissystem** konzipiert und mit der Grundfunktionalität für die definierten Module umgesetzt. Dabei wird eine zielorientierte, explorative Umsetzungsmethodik verfolgt, welche durch fortwährendes Feedback anhand von Anwendungstests und neuen Anforderungen zu modifizieren und zu verfeinern ist. Experimentell können so kontinuierlich neue Ergebnisse und Konzepte untersucht werden und die daraus gewonnenen Erfahrungen in das Gesamtsystem einfließen. Durch evolutionäres Prototyping entsteht auf diese Weise systematisch eine stabile Basis für die Realisierung spezifischer Funktionen, wobei die Anwenderresonanz durch visuelle Zwischenresultate direkt einbezogen wird. Die vorgestellten Konzepte für die Visualisierung der Produktstruktur werden hierbei prototypisch umgesetzt und in die Systemarchitektur eingebunden. Im Fokus steht vor allem die Abbildung des Produktstammbaums anhand der elementaren Ausprägung Aggregation und der Einbezug von Metadaten zu den zugehörigen Bauteilen und Baugruppen. Die beiden entstehenden Prototypen, Kapselung und Knoten-Kanten-Layout stellen dabei unabhängige Anwendungen dar, welche allerdings auf der gleichen Datenstruktur und Grundfunktionalität beruhen. Dies ist insbesondere für den Vergleich und die Bewertung der Layoutkonzepte in der abschließenden Benutzerstudie erforderlich, bei der allgemeine Gebrauchstauglichkeit und die spezifische Expressivität, Effektivität und Angemessenheit der Prototypen unter den Voraussetzungen der Produktstrukturen analysiert werden. Die Einbeziehung der unterschiedlichen Visualisierungskonzepte wird durch das modulare Basissystem ermöglicht, welches die konkrete Datenhaltung und Funktionalität abstrahiert und generalisierte Schnittstellen zur Verfügung stellt. So sind flexible, von konkreten Layoutkonzepten unabhängige Anpassungen der Basissoftware und die Erprobung weiterer Konzepte und deren Kombination mit geringem Implementationsaufwand realisierbar.

Das VIS wird über eine simple **Middleware** mit einem PDM-System gekoppelt, wodurch hohe Flexibilität gewährleistet wird. Mittels einer anwendungsneutralen Schnittstelle kann die Kopplung mit beliebigen PDM-Systemen erfolgen. Auf diese Weise ist PDM-seitig

⁴⁰ Open Source bzw. quelloffen ist eine Ansammlung von Softwarelizenzen, deren Quelltext öffentlich zugänglich ist und durch die Lizenz Weiterentwicklungen fördert.

darüber hinaus nur ein minimaler Anpassungsaufwand erforderlich. Dieses Vorgehen bietet somit eine optimale Abstraktion der Eingangs- und Ausgangsdaten und ermöglicht in fortführenden Entwicklungen das Zurückführen der Daten zum PDM-System oder die Komposition verschiedener Basisdaten. Die visualisierungsrelevanten Daten können direkt über die Schnittstelle abgerufen oder effizient in geeignetem Format abgelegt werden. Durch die Verwendung von implementationsunabhängiger Datenstrukturierung entsteht überdies eine ausbaufähige Datenhaltung und zugleich eine Anbindungsmöglichkeit für andere Systeme.

Als grundlegende Programmiersprache für die Entwicklung der Softwarekomponenten wird die objektorientierte und betriebssystemunabhängige Programmiersprache **Java** eingesetzt, welche im Jahre 1995 erstmalig durch das Unternehmen Sun Microsystems eingeführt wurde. Im Kontext der Java-Technologie werden Java-Programme in Bytecode übersetzt und in einer speziellen Laufzeitumgebung ausgeführt. Die wichtigste Komponente ist dabei die Java Virtual Machine (Java-VM), welche den Bytecode interpretiert und bei Bedarf kompiliert (Just-in-time-Kompilierung). Generell kann davon ausgegangen werden, dass Java-Programme ohne weitere Anpassungen auf verschiedenen Computern und Betriebssystemen lauffähig sind, wenn für diese eine Java-VM existiert. Java hat syntaktische Ähnlichkeit zu C++, wurde aber zugunsten der Einfachheit im Vergleich zu anderen objektorientierten Programmiersprachen im Sprachumfang reduziert. Spezielle Merkmale von Java unterstützen die Robustheit von Anwendungen und verringern somit die Wahrscheinlichkeit von Systemfehlern. Zu den Haupteigenschaften zählen weiterhin eine umfangreiche Klassenbibliothek mit vorgefertigten Komponenten und Entwurfsschablonen sowie Sicherheitsmechanismen und Werkzeuge zur Erstellung verteilter Anwendungen. Letztlich sind gesonderte Java-Programme auch serverseitig als Webservice oder clientseitig als eingebettetes Java-Applet in Webbrowsern integrier- und nutzbar. Idealerweise führt die Verwendung von Java zu dynamischen Programmen, welche den Entwicklungs-, Vertriebs- und Administrationsaufwand minimieren.

Die Standardentwicklungsumgebung von Java ist das Java Development Kit (JDK). Das **JDK** ist eine Sammlung von Werkzeugen für die Entwicklung von Java-Programmen und beinhaltet die benötigte Laufzeitumgebung Java Runtime Environment (JRE) zum Ausführen der Applikationen. Die wichtigsten Bestandteile bilden dabei der Java-Compiler, -Dokumentationswerkzeug, -Archiver und die Java-Klassenbibliothek⁴¹, wodurch die Realisierung eigenständiger oder verteilter Anwendungen möglich wird.

6.1 Realisierung der Integrationsschnittstelle

Die erweiterbare Auszeichnungssprache **XML** (eXtensible Markup Language) dient zur Beschreibung des Inhalts von XML-Dokumentenformaten und wird vorrangig zur

⁴¹ Die Java-Klassenbibliothek wird in diverse Pakete (Packages) unterteilt, welche bestimmten Funktionsgruppen entsprechen.

Abbildung hierarchisch strukturierter Daten in Form von Textdateien genutzt. Die XML-Spezifikation, erstmalig im Jahr 1998 vom World Wide Web Consortium (W3C) herausgegeben, definiert hierfür eine Metasprache mittels derer anwendungsspezifische Sprachen festgelegt werden können. XML ist ein offener und freier Standard, welcher besonders auf Einfachheit, Generalität und Nutzbarkeit der Datenbeschreibung abzielt. Der flexible Austausch und Vertrieb von XML-Daten über Kommunikationsverbünde, wie zum Beispiel das Internet, ist ein weiterer Aspekt des Standards. Allgemeine Zielstellung ist die abstrakte Beschreibung der Daten, um verschiedene Ausgabeformen bei der Repräsentation und Visualisierung zu ermöglichen. Grundsätzlich sind dabei Namen und Kombination der beschreibenden Strukturelemente (XML-Elemente) unter Beachtung der Wohlgeformtheit⁴² frei wählbar. Für den Datenaustausch ist es jedoch von Vorteil, wenn das verwendete XML-Format einer festgelegten Grammatik folgt, welche den beteiligten Systemen zur Orientierung und Interpretation der Daten dient. Zu diesem Zweck werden durch Schemasprachen, wie der Dokumenttypdefinition (DTD) oder XML-Schema-Definition (XSD), besondere strukturelle und inhaltliche Einschränkungen ausgedrückt, mit denen Aufbau und zulässige Elemente bestimmt sind. Ein Empfängersystem ist demnach in der Lage, die Gültigkeit (Validität) eines XML-Dokumentes anhand der Wohlgeformtheit und entsprechender Grammatik zu überprüfen und weiter zu verwerten. Für die Verarbeitung von XML-Dateien verwendete Programme bzw. Programmteile werden als XML-Parser bezeichnet und dienen dem Auslesen, Interpretieren und gegebenenfalls Überprüfen der enthaltenen Daten.

Das Dateiformat **PLMXML** (eXtensible Markup Language for Product Lifecycle Management) wurde von Siemens PLM Software eingeführt und dient dem Austausch von geometriellosen Informationen, von Meta- und Strukturdaten im Zusammenhang mit der Produktentwicklung und -fertigung [Sie10]. Das Format basiert auf der erweiterbaren Auszeichnungssprache XML und ermöglicht daher die Abbildung weitreichender und komplexer Prozessinformationen, Baugruppen, Produktstrukturen und deren Beziehungen in lesbarer Form. Des Weiteren werden zugehörige Dokumente und externe Ressourcen teils direkt oder über Referenzierungsmechanismen eingebunden, so dass PLMXML als leichtgewichtiges und flexibles Transportmedium und als Interoperabilitätsmethode von Produktdaten innerhalb des gesamten Produktlebenszyklus dient. Es handelt sich dabei um ein offenes Format, welches auf XML-Standards des W3C basiert und zusammen mit JT (Jupiter Tessellation)⁴³ einen De-facto-Standard für Produktvisualisierung, Kollaboration und Datenaustausch in der Fertigungstechnik bildet. Analog zu XML ist PLMXML über diverse Kommunikationswege austauschbar, verbindet datenerzeugende und datenverwaltende Systeme und bildet die Integrationsschnittstelle für weitere Anwendungstypen.

⁴² Ein XML-Dokument wird als „wohlgeformt“ bezeichnet wenn es alle grundlegenden XML-Regeln einhält. Die XML-Regeln erfordern unter anderem die Existenz eines Wurzelements, Beginn- und End-Auszeichner (Tags) für alle Elemente mit Inhalt und die ebenentreu-paarige Verschachtelung dieser Tags.

⁴³ JT ist ein kompaktes 3D-Datenformat, für tesselierte Dreiecksflächen-Geometrie und exakte B-Rep-Geometrie.

PDM-Systeme stellen diverse Möglichkeiten für Erzeugung und Distribution von PLMXML-Daten über spezielle Exportfunktionen zur Verfügung. Dazu gehören dynamische Abfragen über Serviceschnittstellen oder lokale Ablage in Dateien. Die PLMXML-Daten können dabei zu jeglichen Produkten in der PDM-Datenbank erstellt werden, wobei der Ausgabeumfang und -inhalt zumeist anhand zusätzlicher Optionen und Regeln festgelegt wird. Neben den PDM-Systemen existieren weitere, datenerzeugende und -verarbeitende Systeme, die analog Produktdaten im PLMXML-Format zur Verfügung stellen. Für die Bereitstellung valider Dokumente werden hierfür PLMXML-Schemata eingesetzt, welche die Produktstruktur und assoziierte Produktdaten in erweiterbarer Form beschreiben und Mechanismen zur Referenzierung externer geometrischer Bauteilrepräsentationen und deren Assoziationen enthalten. Durch verschiedene Datenquellen und Exportmechanismen können die Daten eine uneinheitliche Informationsdichte enthalten. Zur Reduktion der redundanten Datenanteile und Erzeugung eines einheitlichen Schemas ist folglich eine Aufbereitung der Daten notwendig.

Die erweiterbare Transformationssprache **XSLT** (eXtensible Stylesheet Language Transformations), als Untermenge von XSL (eXtensible Stylesheet Language), ist eine Programmiersprache zur Umwandlung von XML-basierten Dokumenten. Dabei dienen XSLT-Programme (XSLT-Stylesheets) zur Definition von Übersetzungsregeln auf Basis der logischen Baumstruktur von XML-Dokumenten. Häufig werden so XML-Dateien in andere XML-Dateien oder unterschiedliche Formate umgewandelt. Das Ausgangsdokument bleibt dabei unverändert und dient lediglich als inhaltliche Referenz. Bei der Transformation können dabei Teilmengen des Originalinhaltes übernommen, nicht relevante Elemente ausgelassen und das Datenlayout umstrukturiert werden. Die XSLT-Stylesheets werden von spezieller Software, den **XSLT-Prozessoren**, eingelesen, die unter Beachtung der Stylesheet-Anweisungen die XML-Dokumente parsen und in das gewünschte Ausgabeformat umwandeln.

Saxon, entwickelt von Michael Kay⁴⁴, ist ein quelloffener und standardisierter XSLT-Prozessor für Java. Durch Stabilität und Plattformunabhängigkeit ist Saxon in viele bestehende Java-Programme integrierbar und unterstützt verschiedene Versionen von XSLT und XML-Schema. Die Kombination von Saxon, XML-Schema und geeigneten XSLT-Stylesheets ermöglicht dadurch einen dynamischen Umgang mit PLMXML-Daten und stellt innerhalb dieser Arbeit die eingesetzte Middleware dar. Geeignete Formatierung und Informationsübernahme werden dabei durch individuell abgestimmte Regeln für die nachfolgenden Schritte präpariert.

Im Rahmen dieser Arbeit werden PLMXML-Dateien als grundlegende Datenquelle für die systemtechnische Realisierung der Visualisierungs- und Interaktionsprototypen genutzt und der Middleware als Eingangsdaten zur Verfügung gestellt. Innerhalb der Middleware werden die Daten unter der Verwendung von angepassten XSLT-Skripten und Saxon zunächst einer ersten Umstrukturierung und Aufbereitung unterzogen. Redundante Daten

⁴⁴ Mitglied des W3C und Mitautor von XSLT 2.0 und weiterer XML-basierender Sprachen.

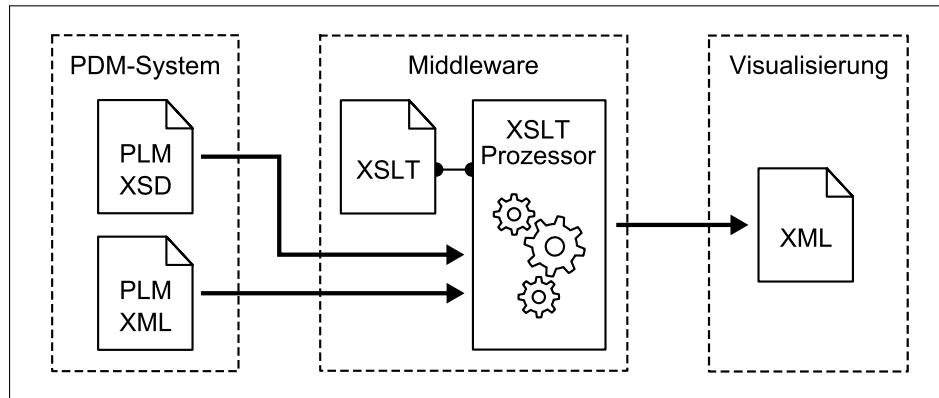


Abbildung 6.1: Architektur der Integrationsschnittstelle mit Ablauf und Funktionseinheiten der XML-Transformation

werden hierbei reduziert und eine angemessene Formatierung für die weiteren Schritte erzeugt. Dabei entstehen effizientere XML-Zwischenformate, welche durch einen XML-Parser in das Basissystem importiert bzw. weiterverarbeitet werden. Die aufbereiteten Daten werden schließlich auf die Datenstruktur des Basissystems abgebildet und für die Visualisierung zur Verfügung gestellt. Durch die Abstraktion der rohen Eingangsdaten wird eine hohe Flexibilität erreicht, wodurch auch weiterhin andere Datenquellen einbezogen werden können. In diesem Fall ist lediglich die Funktionalität der Middleware zu erweitern und auf die Anforderungen weiterer Datendepots abzustimmen, wobei das Visualisierungssystem größtenteils unverändert bleibt.

6.2 Realisierung der Visualisierungskonzepte

Für die Visualisierung der hierarchischen Ordnung einer Produktstruktur, der Aggregation, werden zwei Diagrammkonzepte vorgestellt und deren Implementation sowie Einbindung in das VIS diskutiert. Der Fokus richtet sich dabei speziell auf Diagrammlayouts mit Eignung für sehr komplexe Datenbestände, welche hinreichende Flexibilität aufweisen, um eine große Bandbreite an Strukturarten und unterschiedliche Datenmengen angemessen abbilden zu können. Trotz der sehr unterschiedlichen Darstellungsformen basieren die entwickelten Konzepte auf den gleichen grundlegenden Produktdaten, welche durch die Middleware zur Verfügung gestellt werden. Durch deren generelle Anwendungsfähigkeit auf Daten mit einem hierarchischen Ordnungssystem sind diese allerdings nicht zwangsläufig auf PLMXML-Daten beschränkt. Vielmehr kommen bei der systematischen Entwicklung auch andersartige Datenbestände, wie Dateisysteme oder alternative XML-Daten ohne Produktbezug zum Einsatz, welche zudem weitere Einsatzfelder aufzeigen. Der Schwerpunkt der Arbeit bezieht sich auf die Visualisierung von Produktstrukturen, so dass während des Entwicklungsprozesses und verstärkt in den finalen Phasen entsprechend spezielle Anpassungen vorgenommen werden. Die Entwicklung und Umsetzungsschritte sind in den folgenden Abschnitten beschrieben.

6.2.1 Implementierung des Kapselung-Layouts

Wie bereits im Kapitel 2.6 dargestellt, sind Kapselung-Layouts (Treemaps) eine bekannte Methode zur Abbildung von komplexen und attributiven Daten. Herkömmliche Treemaps beschränken sich dabei allerdings auf rechteckige Formen, welche für die Wahrnehmung der hierarchischen Struktur nur bedingt geeignet sind. Das hier vorgestellte Konzept folgt daher den in [BDL05, BD05] vorgestellten Ansätzen zu Voronoi-Treemaps, welche diese Defizite durch Verwendung und Unterteilung von Polygonen und die Einbettung der Visualisierung in frei wählbare Formen ausgleichen. Das Grundkonzept besteht dabei in der Generierung von Polygon-Treemap-Layouts und der Anwendung von zentrischer Voronoi Tessellation (CVT)⁴⁵. CVT ist eine weitverbreitete Methode zur Energieminimierung mit vielfältigen Anwendungsbereichen, kommt jedoch in der Informationsvisualisierung im Kontext der Voronoi-Treemaps erst seit vergleichsweise kurzer Zeit zum Einsatz.

Ein gebräuchliches Verfahren zur Abbildung von hierarchischen Strukturen ist die Verwendung von Bäumen oder Graphen, welche allerdings mit steigender Elementanzahl allgemein nur bedingt geeignet sind. Die Ursache hierfür ist der wachsende Platzbedarf von Knoten und Kanten und die Limitierung des Anzeigebereichs. **Treemap-Layouts** hingegen unterteilen den zur Verfügung stehenden Bereich auf effiziente Weise, ohne Bildung von Löchern oder Überlagerungen und können ferner spezielle Attribute der Strukturelemente anhand der zugehörigen Flächengröße kodieren. In Abbildung 6.2 wird exemplarisch der Vorgang zur Abbildung der Strukturdaten auf Treemap-Layouts dargestellt. Die Treemap-Layouts entstehen dabei durch rekursive Unterteilung der Begrenzungsfläche anhand der Ausgangsstruktur, so dass jede Teilfläche einem Datenelement entspricht.

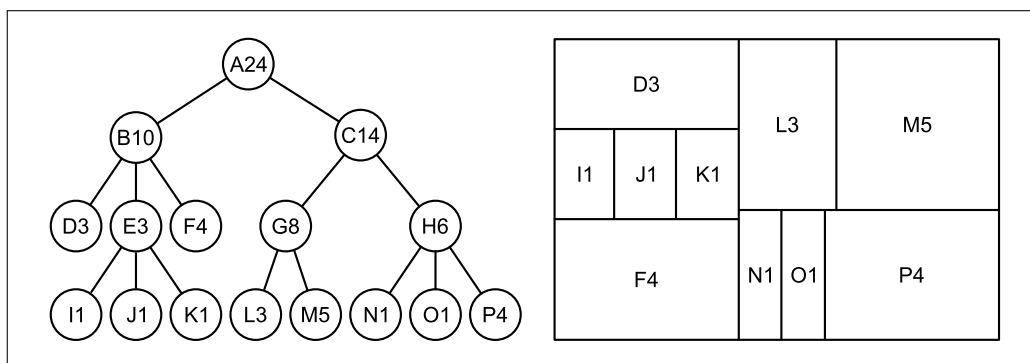


Abbildung 6.2: Prinzipielle Datenabbildung mittels Treemap-Layouts nach [BD05]. Abgebildet ist eine klassische Baumstruktur (links) und ein entsprechendes Treemap-Layout (rechts) der gleichen Ausgangsstruktur. Die Strukturelemente sind jeweils mit einem Namen und einer Gewichtung gekennzeichnet. Die Flächengrößen in der Treemap illustrieren die zugehörigen Elementgewichtungen.

⁴⁵ Centroidal Voronoi tessellations (CVT) ist ein spezieller Typ der Voronoi Tessellation (Voronoi Diagramm), bei dem die Generatorpunkte der Voronoi-Zellen auch gleichzeitig die Schwerpunkt der Zellen sind. CVT entspricht einer optimalen Partitionierung sowie entsprechender Idealverteilung der Generatorpunkte.

Die Voronoi Tessellation (Voronoi Parkettierung) bezeichnet die lückenlose, überlappungsfreie und restlose Überdeckung der euklidischen Ebene durch Parkettsteine bzw. Voronoi-Zellen. Im Rahmen dieser Arbeit werden dabei insbesondere die Aspekte für Treemap-Layouts diskutiert und lediglich planare, zweidimensionale Voronoi-Tessellationen berücksichtigt. Hierfür werden die grundlegenden Definitionen und Eigenschaften nach [BD05] wie folgt beschrieben:

Grundlegende Eigenschaften: $P := \{p_1, \dots, p_n\}$ sei eine Menge n distinkter Punkte in \mathbb{R}^2 mit den Koordinaten $(x_1, y_1), \dots, (x_n, y_n)$. Diese Punkte werden als Generatoren bezeichnet. Weiterhin wird die Unterteilung von \mathbb{R}^2 in n Voronoi Regionen $V(p_i)$ als Voronoi Tessellation $\mathcal{V}(P) := \{V(p_1), \dots, V(p_n)\}$ definiert, wobei die Eigenschaft gilt, dass sich ein Punkt $q(x, y)$ in einer Region $V(p_i)$ befindet, wenn die Voraussetzung $distance(p_i, q) < distance(p_j, q)$ für alle $p_i, p_j \in P$ und $i \neq j$ erfüllt ist. Die Denotation $distance(p_i, q)$ bezeichnet in diesem Zusammenhang eine spezielle Distanzfunktion zwischen dem Generator p_i und dem Punkt q . Allgemein wird die Voronoi Tessellation in einem unbegrenzten Raum definiert. Wird hingegen ein begrenzter Raum S vorausgesetzt, wird von einer begrenzten Voronoi Tessellation $\mathcal{V}_{\cap S}(P) := \{V(p_1) \cap S, \dots, V(p_n) \cap S\}$ P von S gesprochen. Bei einer herkömmlichen Voronoi Tessellation $\mathcal{V}_\varepsilon(P)$ wird die folgende euklidische Metrik als Distanzfunktion verwendet:

$$distance_\varepsilon(p_i, q) := \|p_i - q\| = \sqrt{(x_i - x)^2 + (y_i - y)^2}$$

Der Bisektor von zwei Regionen $V_\varepsilon(p_i)$ und $V_\varepsilon(p_j)$ ist somit die senkrechte Halbierende zwischen den Generatoren p_i und p_j . In Abbildung 6.3 sind begrenzte und unbegrenzte Voronoi Tessellationen beispielhaft abgebildet.

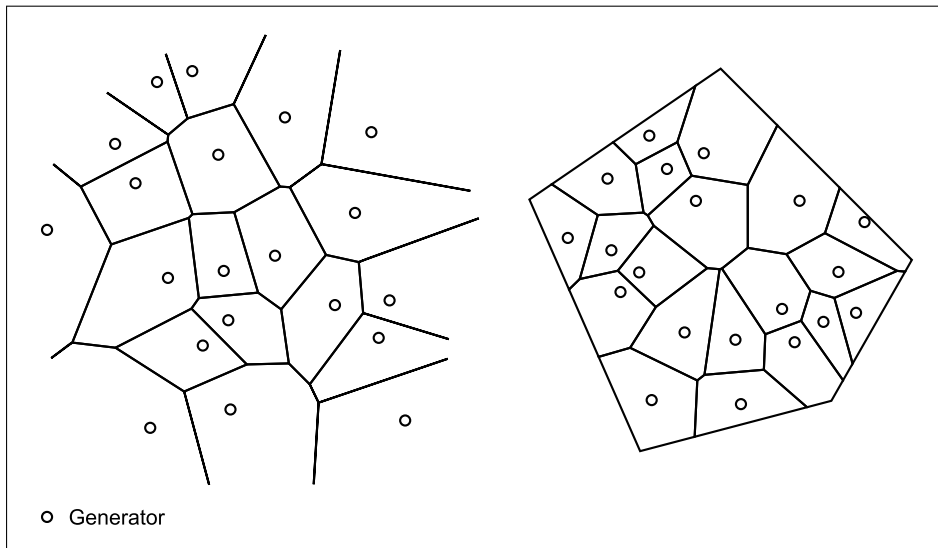


Abbildung 6.3: Beispiele für unbegrenzte und begrenzte Voronoi Tessellation

Gewichtete Voronoi Tessellation: Die herkömmliche Voronoi Tessellation $\mathcal{V}(P)$ geht implizit davon aus, dass jeder Generator die gleiche Gewichtung besitzt. Als Erweiterung hierzu dient eine zusätzliche Parametermenge W , wobei jedem Generator $p_i \in P$ ein

Parameter $w_i \in W$ zugeordnet wird. Diese Parameter werden als Gewichte bezeichnet. Die Parametrisierung der Generatoren ermöglicht die Definition einer gewichteten Distanzfunktion mittels derer die gewichteten Voronoi Tessellation $\mathcal{V}(P, W)$ erzeugt werden kann. In [BD05] werden hierfür die beiden Ausprägungen „additively weighted Voronoi tessellation“ $\mathcal{V}_{aw}(P, W)$ und „additively weighted power Voronoi Tessellations“ $\mathcal{V}_{pw}(P, W)$ vorgestellt, wobei im Rahmen dieser Arbeit lediglich auf die additiv gewichtete Voronoi Tessellation $\mathcal{V}_{aw}(P, W)$ eingegangen wird. Diese verwendet die folgende Distanzfunktion zwischen einem Generator $p_i \in P$ mit zugeordneten Gewicht $w_i \in W$ und einem Punkt q :

$$distance_{aw}(p_i, w_i, q) := \|p_i - q\| - w_i$$

Der Bisektor zwischen den Regionen $V_{aw}(p_i, w_i)$ und $V_{aw}(p_j, w_j)$ formt dabei eine hyperbolische Kurve mit Brennpunkt in p_i und p_j . Die Abbildung 6.4 stellt diesen Zusammenhang beispielhaft dar. Die gewichtete Voronoi Tessellation $\mathcal{V}_{aw}(P, W)$ kann als die Parkettierung mit Kreisen als Generatoren verstanden werden. Der Radius eines Kreises entspricht dabei der Gewichtung des Generators, wobei diese Kreise sich nicht überlagern dürfen.

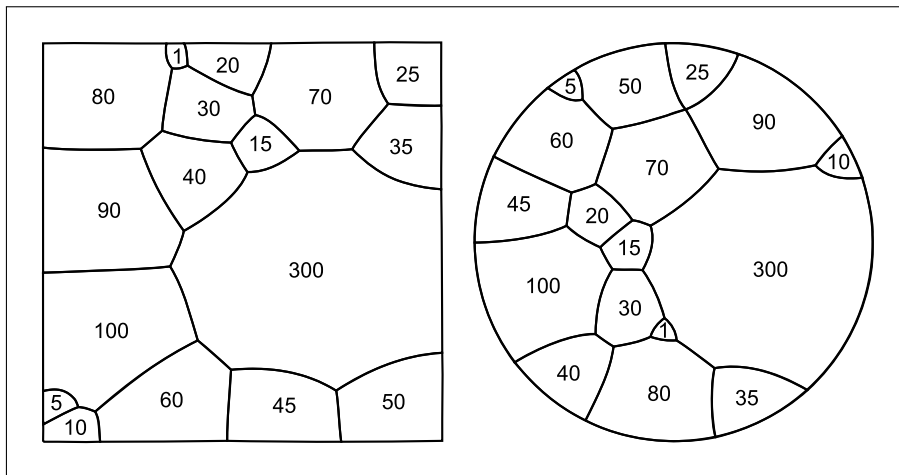


Abbildung 6.4: Beispiele für eine additiv gewichtete Voronoi Tessellation. Die Zahlenwerte in den Voronoi Regionen repräsentieren die Gewichtung der Generatoren.

Zentrische Voronoi Tessellation: Die zentrische Voronoi Tessellation (CVT) besitzt die besondere Eigenschaft, dass jeder Generator p_i dem Schwerpunkt c_i der zugeordneten Voronoi Region $V(p_i)$ entspricht. Der Schwerpunkt einer Voronoi Region $V(p_i)$ wird mit $c_i = \int_{V(p_i)} x dx$ berechnet. Für eine gegebene Menge an Generatoren kann die CVT mit dem iterativen Algorithmus nach Lloyd berechnet werden, wobei von einer initialen Verteilung der Generatoren P im begrenzten Raum S ausgegangen wird. Bei jeder Iteration des Algorithmus wird jeder Generator $p_i \in P$ in den Schwerpunkt c_i der zugeordneten Voronoi Region $V_{\cap S}(p_i) \in \mathcal{V}_{\cap S}(P)$ verschoben. Die Abbruchbedingung für die iterativen Berechnungen besteht darin, dass die Differenz zwischen einem Generator p_i und dem entsprechenden Schwerpunkt c_i unter einem Grenzwert fällt oder eine maximale Anzahl von Iterationen erreicht ist. Das CVT-Konzept kann gleichermaßen auf herkömmliche und gewichtete Voronoi Tessellation angewandt werden, solange der

Schwerpunkt für eine Voronoi Region bestimmt werden kann. Abbildung 6.5 stellt diesen Prozess für eine zufällige Generatorverteilung dar.

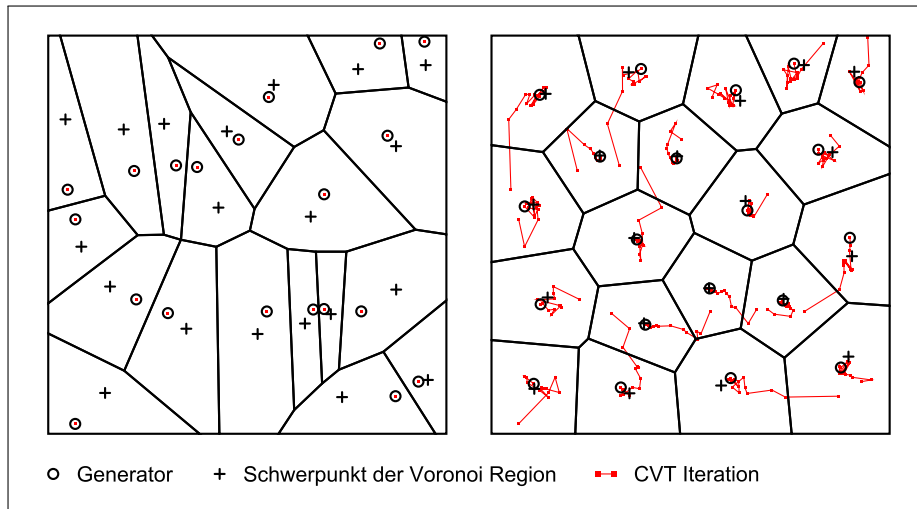


Abbildung 6.5: Beispiele für zentrische Voronoi Tessellation (CVT). Die Darstellung zeigt die Ausgangssituation (links) der Voronoi Tessellation und das entsprechende CVT (rechts) nach 15 Iterationen. Die roten Linien illustrieren zudem die Bewegung der Generatoren während der CVT-Berechnung.

Diskrete Voronoi Tessellation: Die exakte Berechnung von generalisierten Voronoi Tessellationen bzw. gewichteten CVTs ist durch algebraische und kombinatorische Komplexität eine diffizile Problematik. In der Folge werden vermehrt diskrete Annäherungen der Voronoi Tessellation realisiert und eingesetzt. Die Berechnung der diskreten Annäherung einer gewichteten CVT erfolgt durch das Sampling des begrenzten Raumes S mit einer endlichen Punktmenge \tilde{S} und der Errechnung der Zugehörigkeit jedes Samples in \tilde{S} zu einer Voronoi Region. Hierbei können sowohl zufällige, als auch gleichverteilte Samples zum Einsatz kommen [HKL99, RT07, SGM05, JDG02].

Der allgemeine Algorithmus zur Generierung von **Voronoi Treemaps** ist den rekursiven rechteckbasierten Treemap-Algorithmen ähnlich. Die spezielle Anpassung für Voronoi-Treemaps wird hierbei in jedem rekursiven Unterteilungsschritt mittels der CVT-Berechnung durchgeführt. Grundlegend für die Charakteristik ist dabei der Zusammenhang zwischen dem Flächeninhalt der Regionen und der Gewichtung der Elemente in der Hierarchie. Voronoi Regionen herkömmlicher CVTs mit euklidischer Distanzfunktion haben annähernd den gleichen Flächeninhalt und sind daher bedingt oder nur unter bestimmten Voraussetzungen für Treemap-Layouts geeignet. Mittels gewichteter CVTs ist es hingegen möglich, Regionen mit variablem Flächeninhalt zu erzeugen. Zur Kontrolle des Flächeninhalts wird allerdings eine spezielle Anpassung der CVT-Berechnung benötigt. In jeder Iteration wird neben der Aktualisierung der Generatorposition auch der Gewichtsparameter für jeden Generator anhand des momentanen Flächeninhalts der entsprechenden Voronoi Region adaptiv verändert. Durch die fortwährende Anpassung der Generatoreigenschaften kann ein stabiler Zustand erreicht werden, wobei relatives Gewicht der Strukturelemente und relativer Flächeninhalt der entsprechenden Voronoi

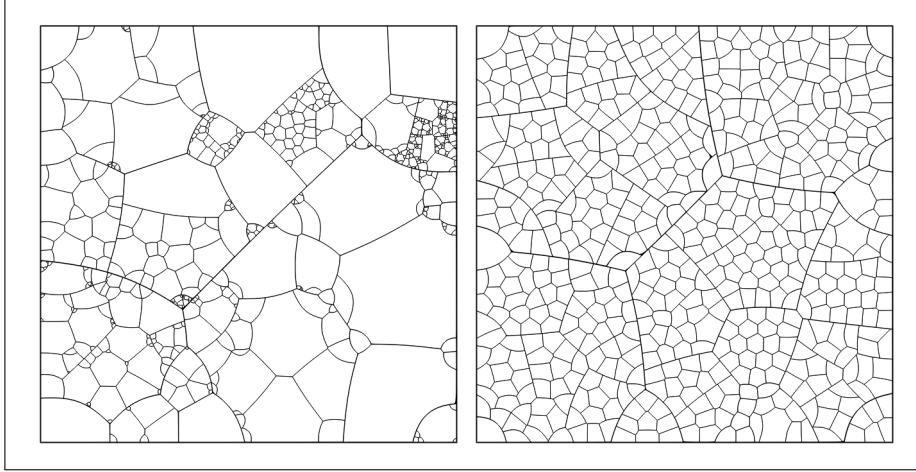


Abbildung 6.6: Beispiele für Voronoi-Treemap-Layouts. Abgebildet sind die additiv gewichtete Voronoi Tessellation mit unterschiedlichen Gewichtungen (links) und gleichen Gewichtsparametern (rechts) des selben Datensatzes mit 675 Elementen auf 6 Hierarchieebenen.

Regionen unter einem maximalen Fehler ε liegen. Der im Rahmen dieser Arbeit verwendete **Algorithmus** orientiert sich an der originalen Beschreibung in [BD05], wobei die gewichtete CVT Berechnung iterativ mittels diskreter Annäherung unter Verwendung von Grafikkhardware (GPU) nach [HKL99] umgesetzt wird. Schwerpunkte und relative Flächeninhalte werden durch Abtasten des Framebuffers ermittelt und zurück an die GPU gesendet, bis eine Konvergenz erreicht ist. Das entsprechende Vorgehen ist in Algorithmus 1 generalisiert und das resultierende Layout wird in Abbildung 6.6 exemplarisch dargestellt.

Algorithm 1 Voronoi-Tessellation

Input: Begrenzter Raum S in \mathbb{R}^2 , geforderte Flächeninhaltsmenge $A_{gef} := \{a_{1_{gef}}, \dots, a_{n_{gef}}\}$ mit $0 < a_{i_{gef}} \leq 1$ und $\sum a_{i_{gef}} = 1$, Schwellwert ε

Output: Unterteilung von S in n disjunkte Teilregionen $s_i \subset S$ mit $\left| \frac{Fläche(s_i)}{Fläche(S)} - a_{i_{gef}} \right| < \varepsilon$

Initialisiere eine Menge von n Punkten $P := \{p_1, \dots, p_n\}$ mit $p_i \in S, p_i \neq p_j$

Initialisiere eine Menge von n Gewichtungen $W := \{w_1, \dots, w_n\}$ mit $w_i = 1$

Initialisiere eine Menge von n Schwerpunkten $C := \{c_1, \dots, c_n\}$

Initialisiere eine Menge von n Flächeninhalten $A := \{a_1, \dots, a_n\}$

Initialisiere eine Datenstruktur für die Voronoi Tessellation $\mathcal{V}_{\cap S}(P, W)$

repeat

$buffer \leftarrow \text{ComputeVoronoiTessellationGPU}(P, W, S)$

$stable = true$

$(C, A) \leftarrow \text{ScanBuffer}(buffer)$

$(stable, P, W) \leftarrow \text{VerifyAndUpdate}(P, C, A, A_{gef}, S, \varepsilon)$

until $stable == true$

$\mathcal{V}_{\cap S}(P, W) \leftarrow \text{ExtractRegions}(buffer)$

Die Aggregation von Produktstrukturen ist durch das hierarchische Ordnungssystem für die Abbildung auf Voronoi-Treemaps geeignet und dient im Kontext dieser Arbeit als Strukturgrundlage für den Generierungsprozess des Layouts. Jede Voronoi Region entspricht dabei einem Bauteil bzw. einer Baugruppe und die Unterteilung einer Region repräsentiert die Elemente einer übergeordneten Klasse. Die allumschließende Begrenzung stellt das Wurzelement dar und die weitere rekursive Parkettierung in verschachtelte Regionen die untergeordneten Elemente. Die räumliche Abgrenzung der Flächen und optimierten Seitenverhältnisse der CVT begünstigen hierbei die Kodierung weiterer relevanter Informationen der Produktstruktur und verschiedener Repräsentationsformen von Bauteilen und Baugruppen. Die Fläche einer Voronoi Region kann beispielsweise für Beschriftungen, bildliche Repräsentation der Bauelemente und Farbkodierung zur Hervorhebung spezieller Struktureinheiten genutzt werden. Weiterhin werden durch die individuellen Formen der Regionen die Differenzierung von Elementen und die Orientierung in der Struktur unterstützt. Die Gewichtung der Generatoren und daraus resultierende relative Zellengrößen können im Zusammenhang mit Produktstrukturen als visuelles Relationsmerkmal von Bauteil- und Baugruppenattributen fungieren. Simple Vergleiche der Elementanzahl von Teilstrukturen bis hin zu speziellen Betrachtungen von kodierten Eigenschaften, wie Entwicklungskosten oder -zeiten von Produktkomponenten können realisiert werden. Somit lassen sich verschiedene spezifische Präsentationsformen einer Struktur generieren, deren Gegenüberstellungen weitere Einblicke und Erkenntnisse ermöglichen. Anhand der Iterationen des rekursiven Generierungsprozesses ist es möglich, gleichmäßige Animationen zu erstellen, mit denen die Layoutentstehung interaktiv verfolgt werden kann. Analog zu anderen Treemap-Layouts sind Weiterentwicklungen, wie Begrenzungslinien, adaptive Breiten der Randlinien, Schattierungen der Regionen (Cushions), etc. auf Voronoi-Treemaps anwendbar. Diese fördern ergänzend die Wahrnehmung und Interpretation des Betrachters, wovon einige in Abbildung 6.7 exemplarisch dargestellt sind.

6.2.2 Implementierung des Knoten-Kanten-Layouts

Die Abbildung von hierarchischen Strukturen mittels Knoten-Kanten-Layouts ist eine konventionelle Methode, wobei die Strukturelemente durch Knoten und deren Beziehungen durch knotenverbindende Kanten repräsentiert werden. Es existieren verschiedenartige Layoutformen, welche sich hauptsächlich durch die räumliche Verteilung der Layoutkomponenten unterscheiden. Gemein ist diesen Methoden, dass unter Verwendung unterschiedlicher Ordnungsmechanismen eine möglichst optimale oder semantisch sinnvolle Distribution der Knoten und Kanten erzielt wird.

Force-Directed (Kräfte-gerichtete) Ansätze sind intuitive Methoden für die Generierung geradliniger Knoten-Kanten-Layouts. Allgemein simulieren Force-Directed Algorithmen ein System von Kräften entsprechend der zugrundeliegenden Datenstruktur. Das Kräftesystem entspricht dabei typischerweise einem physikalischen Modell und ordnet jedem

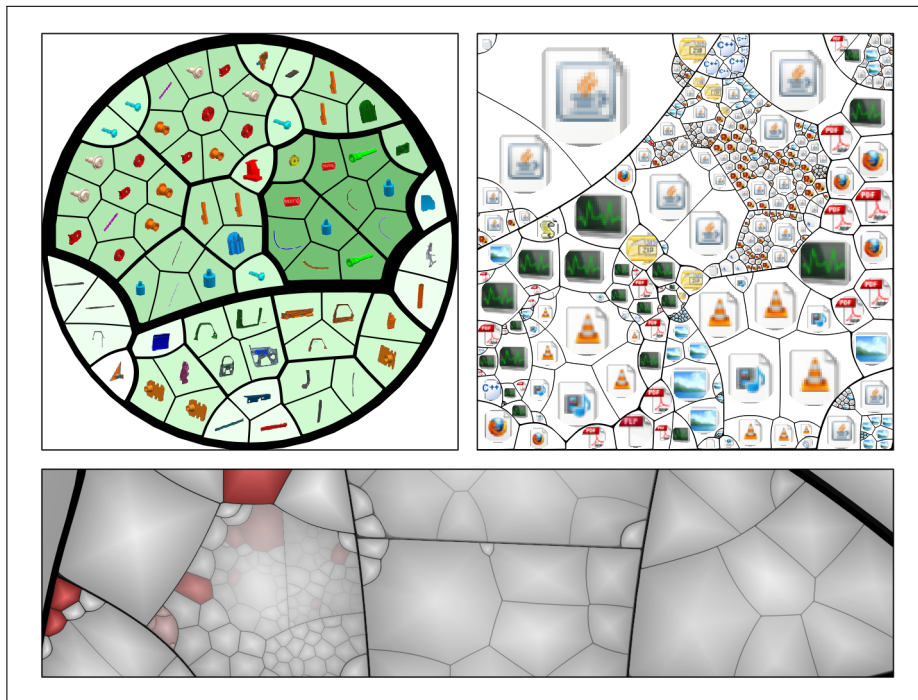


Abbildung 6.7: Experimentelle Darstellungsvarianten von Voronoi-Treemaps. Das erste Bild (links) zeigt die Aggregation einer Produktstruktur mit adaptiven Randlinienbreiten, bildliche Repräsentation der Bauteile und Farbkodierung der Zellen anhand der Strukturtiefe. Das zweite Bild (rechts) ist das Layout eines Dateisystems, wobei die Dateien mittels Icons dargestellt sind und die Zellengrößen dem jeweiligen Speicherbedarf entsprechen. Das dritte Bild (unten) zeigt einen Layoutausschnitt, mit schattierten Zellen, Transparenz anhand der Strukturtiefe und gezielte Hervorhebung spezieller Zellen.

Strukturelement einen Körper und jeder Beziehung eine Kraft zu. In einem iterativen Prozess entsteht daraufhin eine Konfiguration mit minimaler Energie. Jeder Körper befindet sich hierbei in einer Position, in der die Summe aller Kräfte auf den Körper unter einem festgelegten Grenzwert liegt. Force-Directed Methoden besitzen generell folgende Bestandteile:

Das Modell welches das Kräftesystem mit Knoten und Kanten definiert und die physikalischen Eigenschaften des Layouts beschreibt.

Der Algorithmus als Technik zur Generierung der Ausgeglichenheit (**Equilibrium**) des Kräftesystems und einer Knotenverteilung mit minimaler Energie.

Im Rahmen dieser Arbeit wird für die Generierung des Knoten-Kanten-Layouts ein simples Modell mit Federkräften und elektrischen Ladungen verwendet. Die Knoten werden durch sich abstoßende gleichstarke Magneten bzw. Partikel mit gleicher elektrischer Ladung repräsentiert und die zugehörigen Kanten durch knotenverbindende Federkräfte abgebildet. Das verwendete Modell besteht generell aus einer endlichen Knotenmenge $V := \{v_1, \dots, v_n\}$ von n Knoten und einer endlichen Kantenmenge $E := \{e_1, \dots, e_m\}$ von m Kanten zu ungeordneten Knotenpaaren (v_i, v_j) mit $v_i \neq v_j$. Die auf einen Knoten v wirkende Kraft $f(v)$ wird somit wie folgt beschrieben:

$$f(v) = \sum_{(v_i, v_j) \in E} g_{v_i v_j} + \sum_{(v_i, v_j) \in V \times V} h_{v_i v_j}$$

Die erste Komponente $g_{v_i v_j}$ beschreibt die Krafteinwirkung auf einen Knoten v durch die Federwirkung zwischen den Knoten v_i und v_j . Die zweite Komponente $h_{v_i v_j}$ ist die elektrische Abstoßung (**Repulsion**) ausgeübt von Knoten v_i auf v_j . Die Kraft $g_{v_i v_j}$ folgt dem Hookeschen Gesetz, so dass $g_{v_i v_j}$ proportional zur Differenz von der Distanz zwischen v_i und v_j und der Länge der Feder in Gleichgewichtslage (potenzielle Energie = 0) ist. Die elektrische Kraft $h_{v_i v_j}$ folgt dem Abstands- bzw. Entfernungsgesetz ($1/r^2$ -Gesetz).

Ist die euklidische Distanzfunktion zwischen zwei Punkten p und q in \mathbb{R}^2 mit $d_\epsilon(p, q)$ beschrieben und die Position eines Knotens v mit $p_v = (x_v, y_v)$ bestimmt, so ist die x -Komponente der Kraft $f(v)$ auf v wie folgt definiert:

$$\sum_{(v_i, v_j) \in E} k_{v_i v_j}^{(1)} (d_\epsilon(p_{v_i}, p_{v_j}) - l_{v_i v_j}) \frac{x_{v_j} - x_{v_i}}{d_\epsilon(p_{v_i}, p_{v_j})} + \sum_{(v_i, v_j) \in V \times V} \frac{k_{v_i v_j}^{(2)}}{(d_\epsilon(p_{v_i}, p_{v_j}))^2} \frac{x_{v_j} - x_{v_i}}{d_\epsilon(p_{v_i}, p_{v_j})}$$

Die y -Komponente von $f(v)$ folgt in diesem Zusammenhang einem analogen Ausdruck. Die Parameter $l_{v_i v_j}$, $k_{v_i v_j}^{(1)}$ und $k_{v_i v_j}^{(2)}$ sind unabhängig von der Position der Knoten und können wie folgt interpretiert werden:

- Die natürliche Länge einer Feder zwischen v_i und v_j ist $l_{v_i v_j}$. Hat die Feder eine Länge $l_{v_i v_j}$, also $d_\epsilon(p_{v_i}, p_{v_j}) = l_{v_i v_j}$, so befindet sie sich in Gleichgewichtslage und es wirkt keine Federkraft (potenzielle Energie = 0) auf v_i und v_j ein.

- Die Steifigkeit bzw. Konstante einer Feder zwischen v_i und v_j ist durch $k_{v_i v_j}^{(1)}$ beschrieben. Der Wert von $k_{v_i v_j}^{(1)}$ bestimmt die Annäherung der Distanz zwischen v_i und v_j an die Federlänge $l_{v_i v_j}$.
- Die Stärke der elektrischen Repulsion zwischen v_i und v_j richtet sich nach dem Wert von $k_{v_i v_j}^{(2)}$.

Die Eigenschaften des Modells definieren implizit auch ästhetische Kriterien des resultierenden Layouts. Die Federkräfte zwischen den verbundenen Knoten v_i und v_j beeinflussen beispielsweise deren Distanz und sorgen dafür, dass der Abstand in etwa gleich zu $l_{v_i v_j}$ ist. Die elektrischen Kräfte regeln ferner die Abstände zwischen Knoten und verringern Überlappungen. Das entstehende Layout ist folglich stark von den gewählten Parameterwerten abhängig. Durch geeignete Parameterwahl können so symmetrische und individuelle Layoutformen, welche spezielle semantische Struktureigenschaften hervorheben oder Elementattribute implizit kodieren, entstehen.

Der verwendete Prozess zur Erzeugung des Equilibrium folgt einem simplen iterativen Algorithmus. Die Knoten und zugehörigen Beziehungen werden zunächst schrittweise zum Modell hinzugefügt, bis die gesamte Datenstruktur abgebildet ist. Gleichzeitig werden die Knotenpositionen im Modell anhand der Krafteinwirkungen aktualisiert und bereits während der Vervollständigung inkrementell optimiert. Diese Vorgehensweise vermindert Überlappungen von Knoten und sich überschneidende Kanten. Der Generierungsprozess wird anschließend solange fortgesetzt, bis die Energie des Kräftesystems unter einen Grenzwert fällt oder eine maximale Anzahl an Iterationen erreicht ist. Ein generalisierter Ablauf ist in Algorithmus 2 beschrieben und die resultierenden Iterationsschritte sind in Abbildung 6.8 beispielhaft dargestellt.

Algorithm 2 Force-Directed-Layout Generierung

Input: Menge von n Knoten $V := \{v_1, \dots, v_n\}$, Menge von m Kanten $E := \{e_1, \dots, e_m\}$, Menge von k Parametern $C := \{c_1, \dots, c_k\}$, Schwellwert ε

Output: Positionen der Knoten im Equilibrium mit $\sum_{i=1}^n f(v_i) < \varepsilon$

Initialisiere eine leere Menge von Punkten $P := \emptyset$ mit Kapazität n

Initialisiere eine leere Menge von Kräften $F := \emptyset$ mit Kapazität n

repeat

$(nodesRemaining, P, F) \leftarrow \text{addNextNodeToModel}(V, E, C)$

$stable = true$

$F \leftarrow \text{CalculateAttraction}(P, E, C)$

$F \leftarrow \text{CalculateRepulsion}(P, V, C)$

$(stable, P) \leftarrow \text{VerifyAndUpdate}(P, V, E, C, F, \varepsilon)$

until $nodesRemaining == false$ **and** $stable == true$

$V \leftarrow \text{SetNodePositions}(P, V)$

Im Kontext dieser Arbeit dient die Aggregation von Produktstrukturen als grundlegende Datenstruktur für die Generierung des Force-Directed-Layouts. Bauteile und Baugruppen entsprechen den Knoten des Layouts und deren hierarchische Beziehungen werden

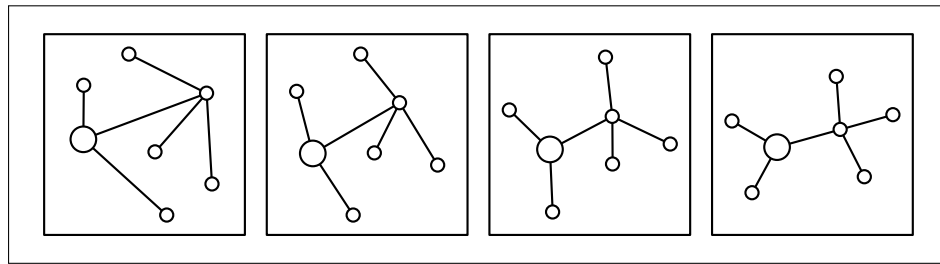


Abbildung 6.8: Generierungsprozess des Knoten-Kanten-Layouts anhand einer Force-Directed Methode. Die Darstellung zeigt die Ausgangssituation (links) des Layouts, zwei Zwischenkonfiguration (mitte) und das resultierende Equilibrium (rechts).

durch Kanten repräsentiert. Die Knoten selbst können beispielsweise durch geometrische Formen (Kreis, Rechteck, Polygon, etc.) dargestellt und die Beziehungen mittels gerader Linien veranschaulicht werden. Die Flächen der Knoten und die Umgebung der Linien können für zusätzliche Kodierung weiterer schriftlicher oder bildlicher Informationen, verschiedener Repräsentationsformen oder Metadaten genutzt werden. Die Farbkodierung und die Variation von Größe und Form der Knoten, kann weiterhin zur Hervorhebung spezifischer Details, Illustration von Bauteil- und Baugruppenattributen und für die Orientierung im Layout eingesetzt werden. Die Parametrisierung der Modelleigenschaften und der individuellen Bedingungen ermöglichen den Einbezug von speziellen Konfigurationskriterien. Beispielsweise können so anhand der Kantenlänge Elementanzahlen von Teilstrukturen abgelesen oder betonte Knoten leichter identifiziert werden. Analog zum Kapselung-Layout ist es möglich, unterschiedliche Präsentationsformen einer Struktur zu erzeugen, anhand des Generierungsprozesses gleichmäßige Animationen zu erstellen und weitere visuelle Anpassungen vorzunehmen. In Abbildung 6.9 sind einige experimentelle Darstellungsvarianten des Force-Directed-Layout abgebildet.

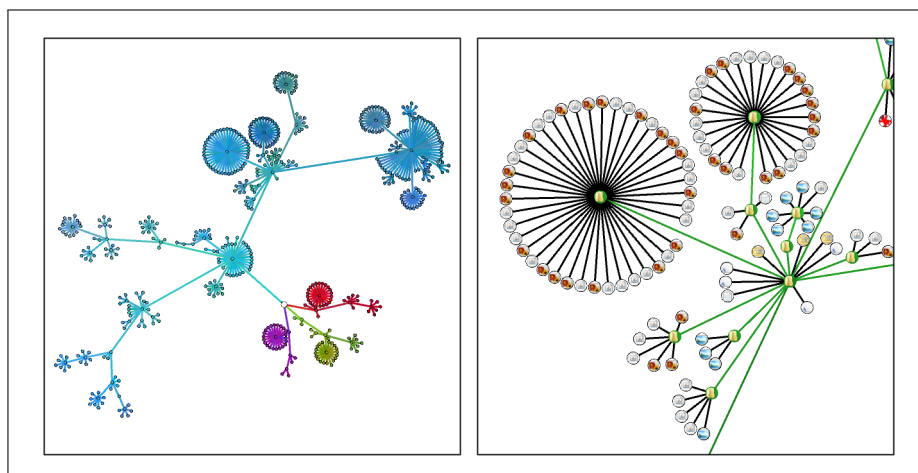


Abbildung 6.9: Experimentelle Darstellungsvarianten von Force-Directed-Layouts. Das erste Bild (links) zeigt eine Struktur mit 534 Elementen auf 8 Hierarchieebenen und Farbkodierung anhand von Elementkategorien. Das zweite Bild (rechts) zeigt den Layoutausschnitt eines Dateisystems und die Darstellung der Dateien mittels Icons.

6.3 Realisierung der Datenhaltung

Die Basis der **Datenhaltung** für das VIS bilden die bereits in Abschnitt 6.1 vorgestellten Werkzeuge der XML-Sprachfamilie. Die prototypische Realisierung umfasst die Abbildung der hierarchischen Struktur der Aggregation, der bauteil- und baugruppenbezogenen Informationen, die Generierungs- und Layoutdaten der Visualisierungskonzepte und weiterer Anwendungsparameter und Benutzereinstellungen. Generierte Layouts und Benutzermodifikationen können in individuellen XML-Dateien abgespeichert werden. Das XML-Format fördert die Erweiterung der Datenhaltung und bietet genügend Flexibilität, um jegliche Produktinformationen oder anwendungsbezogene Daten aufzunehmen. Die persistente Speicherung der Layoutkomponenten ermöglicht den Abruf und die direkte Visualisierung bestehender Strukturlayouts, ohne diese erneut generieren zu müssen. Verschiedene Änderungsstände und Versionen einer Struktur oder unterschiedlich kodierte Layouts lassen sich so erzeugen und zu einem anderen Zeitpunkt mit minimalem Zeitaufwand analysieren und vergleichen.

Das Datenmanagement der einzelnen XML-Dateien und der zugehörigen Ressourcen erfordert entsprechend dem Projektumfang weitere Verwaltungsstrategien und -mechanismen. Für die im Rahmen der Arbeit entstehenden Projekte ist ein Ordnungsschema mittels festgelegter Dateisystemstrukturen ausreichend, welche die Daten anwendungs- global oder lokal für ein konkretes Projekt organisieren. Für eine weiterführende Administration lassen sich übergeordnete relationale Datenbanken und entsprechende DBMS mit geringem Aufwand integrieren. Die **Layoutdateien** enthalten neben der Aggregationsstruktur und grundlegenden Visualisierungskomponenten Verweise auf weitere externe Ressourcen und Daten. Auf diese Weise wird eine einfache und effiziente Daten- und Speicherverwaltung erreicht. Beispielsweise können spezielle Daten so zentral abgelegt, von mehreren Layouts genutzt und über Kommunikationswege abgerufen werden. In Tabelle 6.1 ist der prinzipielle Aufbau der Layout-XML-Dateien ausschnittsweise mit exemplarischen Inhalt dargestellt.

Die Strukturierung des Inhalts der Layout-XML-Dateien folgt der Aggregation der Produktstruktur und gliedert die `<Node>`-Elemente entsprechend der hierarchischen Beziehungen von Bauteilen und Baugruppen. Die Tags in Zeile 3–8 beziehen sich auf die Visualisierungseigenschaften und das `<Data>`-Element enthält zusätzliche Produktdaten und Verweise auf externe Ressourcen für das zugehörige `<Node>`-Element.

6.4 Realisierung der Benutzungsschnittstelle und Interaktionsgestaltung

Die Realisierung des Interaction Design für das VIS orientiert sich an den vorgestellten Grundlagen, Anforderungen und Richtlinien für benutzerorientierte und interaktive Systeme. Das grundlegende Konzept für die Benutzungsschnittstelle und Interaktion mit den

```

1  <Graph date="2010:12:16:15:12:04">
2    <Node>
3      <Position>0.5:0.5</Position>
4      <GSize>3.87264096E8</GSize>
5      <Color>254:255:255</Color>
6      <Bounds>0.0:0.0;1.0:0.0;1.0:1.0;0.0:1.0;1.0:1.0</Bounds>
7      <ICircle>0.5:0.5:0.498972696</ICircle>
8      <Poly>0.0:0.0;0.0:1.0;1.0:1.0;1.0:0.0;0.5:0.0</Poly>
9      <Data>
10       <Attr><Key>id</Key><Val>000483</Val></Attr>
11       <Attr><Key>name</Key><Val>Front Door</Val></Attr>
12       <Attr><Key>image</Key><Val>CD340_ASM\pv_FRTDR.jpg</Val></Attr>
13       <Attr><Key>type</Key><Val>assembly</Val></Attr>
14       <Attr><Key>weight</Key><Val>94.0</Val></Attr>
15     </Data>
16
17     <Node> <Node> ... </Node> ... </Node>
18     ...
19   </Node>
20 </Graph>

```

Tabelle 6.1: Allgemeiner Aufbau der Datenhaltung im XML-Format

generierten Visualisierungslayouts ist eine **skalierbare Benutzungsoberfläche (ZUI)**. Wie bereits in Kapitel 2.7 beschrieben, ermöglicht das Prinzip einer ZUI die Vergrößerung, Verkleinerung und Verschiebung des Anzeigebereichs und folglich die Erkundung des generierten Layouts. Des Weiteren kann die Präsentation um einen definierbaren Punkt gedreht werden, um individuelle Ausrichtungen des Layouts zu erzeugen. Diese zusätzliche Funktionalität ist insbesondere für die Eigenschaften spezieller Präsentationsmedien unterstützend und erlaubt die Justierung der Darstellung an individuelle Blickwinkel der Benutzer. Die Modifikation des Skalierungsfaktors und das Panning und Rotieren des Darstellungsfeldes, wird interaktiv und stufenlos durch den Benutzer gesteuert. Auf diese Weise können zu jeder Zeit Details und Übersichten explorativ untersucht werden. Dies ist vor allem bei sehr komplexen Datenstrukturen und daraus resultierenden Layouts von Vorteil, welche sich anhand der verschiedenen Visualisierungskomponenten ähnlich einer Landkarte analysieren lassen. Gezieltes Zoomen vergrößert oder verkleinert die Darstellung und behält gleichzeitig den Skalierungsfokus im Anzeigebereich. Die prototypische Umsetzung nutzt dabei eine geometrische Skalierung, die alle Darstellungskomponenten einheitlich und mit adaptiver Auflösung anpasst. Lediglich für den Betrachter nicht sichtbare Komponenten, wie zum Beispiel durch Verdeckungen, nicht wahrnehmbare Größen oder Positionen außerhalb des Anzeigebereichs, werden zugunsten der Performance beim Rendering ausgelassen. In Abbildung 6.10 ist das verwendete Konzept schematisch beschrieben.

Die Präsentation und Navigation der Visualisierungslayouts ist neben der grundlegenden ZUI Funktionalität in zwei Modi unterteilt, welche für den Benutzer während der

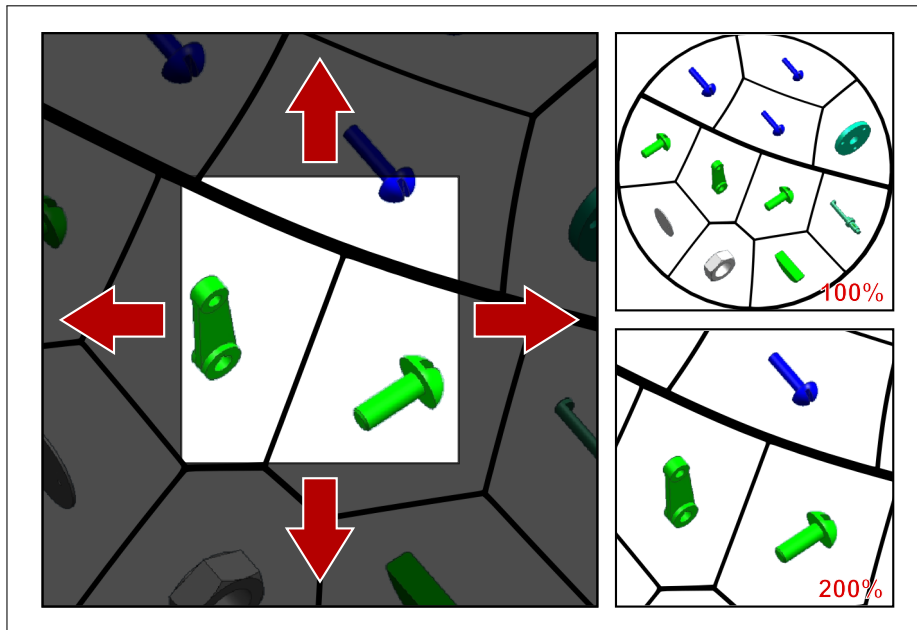


Abbildung 6.10: Grundlegendes Konzept der skalierbaren Benutzungsschnittstelle (ZUI)

Analyse frei wählbar und umschaltbar sind. Der Übersichtsmodus und der Explorationsmodus unterscheiden sich primär in der Darstellung und Selektion der grafischen Layoutkomponenten. Dies ist erforderlich, um die speziellen Eigenschaften der Layoutkonzepte anzugleichen und den Benutzer bei der Bedienung und Untersuchung verschiedenartiger Datenstrukturen zu unterstützen. Der **Übersichtsmodus** stellt ein Layout in seiner Gesamtheit dar und ermöglicht unter anderem die Betrachtung und Auswertung der kompletten Struktur, übergeordneter Zusammenhänge und von Teilbereichen und Kontext. Aufgrund des fundamentalen Layoutprinzips von Treemaps und unter bestimmten Voraussetzungen, ist dieser Modus allerdings für eine effiziente Visualisierung nicht ausreichend. In der prototypischen Umsetzung werden im Übersichtsmodus bei Treemap-Layouts nur bauteilbezogene Komponenten abgebildet. Folglich können diese ausschließlich durch Farbkodierung und Begrenzungslinien oder andere Techniken den Baugruppen zugeordnet werden. Die gesonderte Abbildung von Baugruppen ist in diesem Fall allerdings nicht realisierbar. Auch ist der Übersichtsmodus unter Umständen nur bedingt geeignet. Hierzu zählen beispielsweise sehr große und komplexe Datenstrukturen, gezielte Abfragen bestimmter Details, Selektion einzelner Elemente und die Orientierung bei starker Vergrößerung des Layouts. Der **Explorationsmodus** gestattet hingegen eine kontinuierliche Navigation durch die Layoutstruktur und die Erzeugung eines Navigationspfades durch inkrementelle Selektionen von Strukturelementen. In der Bilderzeugung werden folglich nur Layoutkomponenten berücksichtigt, welche durch den Navigationspfad bestimmt sind oder direkte Verbindungen zu Navigationselementen besitzen. Im Explorationsmodus beginnt die Navigation in der Regel am Wurzelement und wird schrittweise entlang der Strukturhierarchie verfeinert oder abstrahiert. In Abbildung 6.11 ist diese Vorgehensweise beispielhaft dargestellt. Auf diese Weise werden eine effiziente Vorgehensweise, element- und navigationspfadbezogene Analysen und

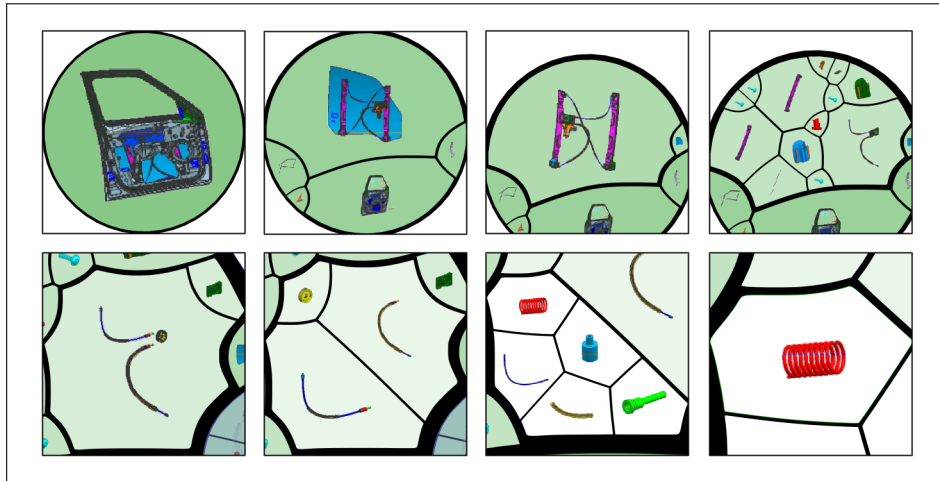


Abbildung 6.11: Der Explorationsmodus der Benutzungsschnittstelle. Abgebildet ist die Navigation durch ein Kapselung-Layout mit automatischer Zoomfunktion.

die Reduktion der visuellen Informationsmenge erreicht. Die explorative Untersuchung der Elemente richtet sich dabei nach einem definierbaren geradlinigen Ablauf oder folgt den weiterführenden Abzweigungen entlang des Navigationspfades. Die zusätzliche Hervorhebung des Navigationspfades im Übersichtsmodus unterstützt die Betrachtung der Selektion und des Vorgehens im Kontext, wobei gleichzeitig ein neues unabhängiges Startelement für die Navigation im Explorationsmodus ausgewählt werden kann.

Die Präsentation und Auswahl von Strukturelementen, die Abfrage von zugehörigen Informationen und die Bedienung der GUI-Elemente folgt im Übersichts- und Explorationsmodus dem gleichen Schema. Ausschlaggebend für Abbildung und Interaktion mit den grafischen Komponenten ist vorrangig die Einbindung in einen **Szenengraph**⁴⁶ mit einer Bounding-Volume-Hierarchie. Der Szenengraph ermöglicht die effiziente hierarchische Modellierung und Manipulation der Objekte in der darzustellenden Szene. Jedem Objekt in der Szenenhierarchie werden bestimmte Eigenschaften und Attribute zugeordnet. Die Änderungen an einem Objekt kann sich dabei potenziell auf alle untergeordneten Objekte auswirken. In diesem Zusammenhang werden Objekteigenschaften und Global- bzw. Welteigenschaften unterschieden. Objekteigenschaften beziehen sich auf die Eigenschaften eines Objekts in Abhängigkeit des übergeordneten Objektes. Welteigenschaften beschreiben hingegen die Eigenschaften eines Objekts bezüglich der gesamten Szene. Die Handhabung der Szene und der zugeordneten grafischen Bestandteile wird deutlich vereinfacht, da übergeordnete Modifikationen, wie beispielsweise die Transformation der Szene, sich direkt auf die Gesamtheit der betreffenden Objekte beziehen. Die Bounding-Volume-Hierarchie ordnet jedem Objekt ein Bounding-Volume⁴⁷ zu, welches die räumliche Ausdehnung des Objekts einschließlich dessen untergeord-

⁴⁶ Ein Szenengraph ist eine zumeist hierarchische und objektorientierte Datenstruktur, mit der die logische und räumliche Anordnung von darzustellenden Elementen in Szenen beschrieben wird.

⁴⁷ Ein Bounding-Volume bzw. eine Bounding-Box ist ein einfacher geometrischer Körper, wie Rechteck, Kreis (2D) oder Quader, Sphäre (3D), der ein komplexes Objekt umschließt.

neten Objekten definiert. Die strukturelle Gliederung der Bounding-Volumes erleichtert den Umgang mit den Visualisierungskomponenten und unterstützt effizientes Rendering, Attributabfragen und Berechnungen wie Kollisionsabfragen. Bei der Bilderzeugung und Auswahl werden nur Objekte berücksichtigt, deren übergeordnete Objekte bereits als zugehörig klassifiziert wurden. Die **Darstellung** und **Auswahl** der Bedien- und Layoutkomponenten richtet sich ebenso nach der Ordnungsstruktur des Szenengraph. Die Abbildung der einzelnen Komponenten folgt der Szenenhierarchie und so werden die Bestandteile mit dem Wurzelement beginnend schrittweise bis zum Hierarchieende in die Bildgenerierung einbezogen. Auf diese Weise liegen übergeordnete Objekte im Szenenhintergrund und werden gegebenenfalls von untergeordneten Objekten teilweise verdeckt. Ferner wird dadurch ein effizientes Auswahlverfahren erzeugt, welches sich bezüglich des Auswahlpunktes immer auf das am weitesten im Vordergrund befindliche Objekt bezieht.

Die **Selektion** und **Aktivierung** der grafischen Bedienbestandteile ist durch deren dargestellte Form bedingt. Befindet sich ein Eingabepunkt innerhalb der abgebildeten Fläche einer Komponente im Darstellungsraum, so können mit dieser Interaktionen durchgeführt werden. Im Kontext der Visualisierungslayouts betrifft dies vor allem die Aggregationselemente repräsentierenden Knoten und Voronoi-Regionen, mit denen der Benutzer Aktionen ausführen kann. Die Bewegung eines Eingabepunktes in die bzw. aus den Bounding-Volumes der Layoutelemente ändert deren Zustand und führt zur Aktivierung oder Deaktivierung der Elemente. Die Aktivierung eines Elements hebt, in Abhängigkeit vom Präsentationsmodus, dessen strukturelle Verbindungen und die zugehörigen direkt über- und untergeordneten Elemente hervor. Diese Funktionalität dient der Analyse des Elementkontextes und im Explorationsmodus zur Vorschau der nächsten Navigationsmöglichkeiten. Mit der Elementaktivierung können weitere elementspezifische Aktionen ausgeführt werden. Der Benutzer hat die Möglichkeit über bestimmte Befehle zusätzliche Informationen nach dem Details-On-Demand-Prinzip zum gewünschten Objekt aufzurufen und dieses zu selektieren. Die Selektion fügt die Elemente zur Auswahlhistorie und zum Navigationspfad hinzu und aktualisiert deren Status. Im Explorationsmodus werden folglich die nächsten Navigationsmöglichkeiten ebenfalls anwählbar und die Navigation und Informationsabfrage kann mit diesen fortgesetzt werden.

Um den Benutzer bei der Interaktion mit dem Visualisierungssystem zu unterstützen, stehen diverse **Automatisierungen** und **Animationen** für gleichmäßige Übergänge zwischen den Zuständen eines Layouts zur Verfügung. Die Navigationsfunktionalität umfasst eine automatisierte Vergrößerung und Zentrierung ausgewählter Komponenten. Das Layout wird bezüglich dem Bounding-Volume des selektierten Elements über eine kurze Zeitspanne kontinuierlich vergrößert, bis der Anzeigebereich den Ausmaßen der Fläche des Elements entspricht. Die gleichzeitige Zentrierung weist dem Element den optimalen Darstellungsraum zu, um genügend Flexibilität für die Abbildung zusätzlicher elementspezifischer Informationen und nächster Navigationsmöglichkeiten zu gewährleisten. Im Explorationsmodus entsteht so eine effiziente Navigation ohne manuelle Zoomaktionen,

welche durch simple Befehle fortgeführt und verfeinert werden kann. Die Navigations- und Verbindungsvorschau bei der Aktivierung eines Elements wird ebenfalls mittels einer ausgewogenen Animation realisiert. Die zugehörigen Elemente und Verbindungen werden in einem stetigen Prozess entsprechend dem angezeigten Layout sichtbar gemacht und bei der Deaktivierung zurück in die Transparenz überführt. Der Benutzer kann die in Abschnitt 6.2 angeführten Generierungsprozesse der Layoutkonzepte interaktiv verfolgen und so Einfluss und Auswirkungen der Elementattribute und Parametrisierung überwachen. In Abbildung 6.12 sind einige dieser Funktionen dargestellt.

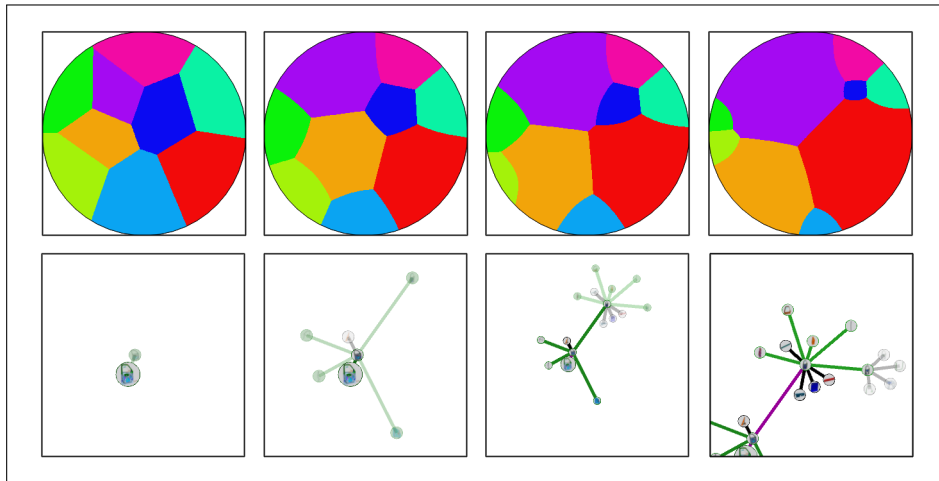


Abbildung 6.12: Automatisierung und Animationen der Benutzungsschnittstelle. Abgebildet sind die Generierungsanimation im Kapselung-Konzept (oben) und Verbindungs- und Navigationsvorschau im Knoten-Kanten-Konzept (unten).

Die Knoten bzw. Voronoi-Regionen werden mit entsprechenden Ikonen und Farbkodierungen dargestellt, um das Wahrnehmungspotenzial der Benutzer auszunutzen und die Differenzierung und Identifikation von Layoutelementen zu unterstützen. Im Kontext der Produktstrukturen entsprechen die **Ikone** den realen Bauteilen und Baugruppen und stellen für dem Benutzer einen direkten Zusammenhang zum Geometriemodell her. Dies erhöht die Wiederauffindbarkeit von Bestandteilen, schafft eine Verbindung zwischen Geometriemodell und Produktstruktur und ermöglicht entsprechend der Produktkenntnis direkte Schlussfolgerungen zu Anordnung, Lage, Kontext und Relationen des Elements in der Struktur. Die Ikone werden automatisiert anhand der Geometriemodelle erstellt und in Form von Bilddaten als externe Ressourcen in die Datenaufbereitung und Datenhaltung einbezogen. Sie repräsentieren eine definierbare Perspektive auf die Bauteile und Baugruppen im 3D-Raum, so dass für jedes Element ein optimales Abbild gewählt werden kann. Die **Farbkodierung** der Layoutkomponenten erfüllt unterschiedliche Funktionen und ist durch den Benutzer einstellbar. Die prototypische Umsetzung des VIS umfasst generell die farbliche Hervorhebung von Elementen und Beziehungen anhand des strukturellen Zusammenhangs, nach gesonderten Vorgaben und die Farbkodierung von Aktivierung, Selektion und Navigationspfad. Beispielsweise werden die Elemente zur Unterstützung der Orientierung mittels der Strukturtiefe, Zugehörigkeit zu einer Teilstruktur oder nach Attributwerten farblich markiert.



Abbildung 6.13: Die grafische Benutzungsoberfläche und Bedienelemente. Abgebildet sind die GUI (unten links) mit den vier Schaltflächen Systeminformationen (oben links), Reset (unten links), Präsentationsmodus (mitte) und Auswahlhistorie (rechts) sowie zwei elementbezogene Informationsfenster, welche die Position der zugeordneten Elemente im Layout anzeigen.

Die grafische Benutzungsoberfläche und deren Bedienelemente werden zugunsten einer simplen Benutzerführung und Förderung des Benutzerfokus auf die wesentliche Layout- und Informationsdarstellung im Umfang auf relevante Bestandteile⁴⁸ reduziert. Wie in Abbildung 6.13 unter anderem dargestellt, umfasst die GUI des VIS vier **Schaltflächen**, welche folgende Funktionalität besitzen:

Layout- und Systeminformationen werden mit Betätigung der entsprechenden Schaltfläche in einem gesonderten Informationsfenster aufgerufen. Dargestellt werden Beschreibungen zur Systembedienung, Informationen zum Fortschritt des Generierungsprozesses, allgemeine Strukturdaten und Layoutdetails.

Reset stellt den Ausgangszustand des Layouts her. Es werden alle Aktivierungen und Selektionen zurückgesetzt und das Wurzelement fokussiert und zentriert.

Umschalten des Präsentationsmodus wählt entsprechend den Übersichts- oder Explorationsmodus für die Visualisierung und Interaktion mit den Layoutkomponenten.

Zurück in der Auswahlhistorie ermöglicht das Aufsteigen in der Strukturhierarchie, wobei jeweils das übergeordnete Element der letzten Selektion aktiviert und selektiert wird.

Die **Informationsfenster** für Layout- und Systeminformationen und die Anzeige von zusätzlichen Details zu Strukturelementen werden unabhängig voneinander und von der Layoutdarstellung bedient. Analog zur Layoutinteraktion können die Fenster skaliert, verschoben, gedreht und von den Anwendern nach Bedarf aufgerufen und geschlossen werden. Die Elementfenster visualisieren fortwährend die aktuelle Position des zugehörigen Elements und erleichtern so Vergleichsoperationen, die Wiederauffindung der

⁴⁸ Die prototypische Umsetzung des VIS erlaubt die weitere Parametrisierung des Systems und Aktions- und Befehlsfolgen mittels Tastatureingaben. Die hierfür benötigten Befehle können über die Systeminformation und -einstellungen abgerufen werden.

Elemente bei Layouttransformationen und die Zuordnung der Fenster. Der Fensterbereich ist inhaltlich anpassbar und bietet die Möglichkeit, weitere elementspezifische Bedien- und Aktionselemente darzustellen. In der prototypischen Umsetzung des VIS umfasst der Fensterinhalt Elementdetails und -ikone.

6.5 Erweiterungen mittels Multi-Touch Technologie

Der Umfang und die Funktionalität des VIS wird im Kontext der Entwicklung eines Demonstrators mit Multi-Touch Technologie erweitert. Der Fokus der Erweiterung richtet sich vorrangig auf die mit dem Einsatz von Multi-Touch verbundenen Möglichkeiten und Potenziale. Schwerpunktmäßig werden die Eignung, Integration und Anwendungsfelder mit dem VIS untersucht. Die grundlegende Funktionalität des Demonstrators wird dabei experimentell umgesetzt und die gewonnenen Ergebnisse fortwährend analysiert und bewertet. Im Demonstrator bezieht sich der Einsatz der Multi-Touch Technologie vor allem auf die Entwicklung von Interaktionstechniken mit den Layoutbestandteilen und Bedienelementen der GUI.

Die Multi-Touch Funktionalität des Demonstrators basiert auf dem quelloffenen und plattformunabhängigen Java Framework **MT4J** [LRZ10], welches nach [KFKW10] ausgewählt wurde. MT4J (Multitouch for Java) ist ein Programmiergerüst mit umfangreichen Funktionen und vorgefertigten Komponenten. Die Entwicklung visueller Anwendungen, die Verarbeitung von multiplen Eingabepunkten und -techniken und der Einsatz auf verschiedenen Präsentationsmedien wird dabei unterstützt. In der prototypischen Umsetzung nutzt der Demonstrator einen spezifischen Teil der MT4J Funktionalität und der generellen Multi-Touch Technologie. Die in Abschnitt 6.4 vorgestellten Interaktionsmethoden für Visualisierungslayout, Bedienelementen der GUI und Informationsfenster werden mittels intuitiver **Interaktionstechniken** erweitert und auf Gesten abgebildet. Der Benutzer kann durch Berührung, in der Regel mit den Fingerspitzen, der Oberfläche eines Touchscreens bestimmte Befehle und Aktionen durchführen. Für die Realisierung der notwendigen Prozesse werden hierfür die grundlegenden Eingabeprozessoren von MT4J genutzt und an die Anforderungen und Eigenschaften des VIS angepasst. Die Eingabeprozessoren erzeugen aus den unverarbeiteten Eingabepunkten übergeordnete semantische Gesten-Events, welche für bestimmte Benutzerbefehle verwendet werden. Die Gesten beschreiben simple Aktionen, wie das kurzzeitige Berühren des Touchscreen, bis hin zu komplexen Aktionsfolgen unter Einbezug multipler Berührungspunkte. In Abbildung 6.14 sind die implementierten Gesten illustriert, mit denen folgende Befehle ausgeführt werden können:

Pan wird durch die Berührung mit zwei Fingern ausgeführt und verschiebt das gesamte Layout entsprechend der Eingabepunkte fortwährend in alle Richtungen. Als Bedingung müssen die Eingabepunkte nach der Berührung in etwa die gleiche Distanz behalten.

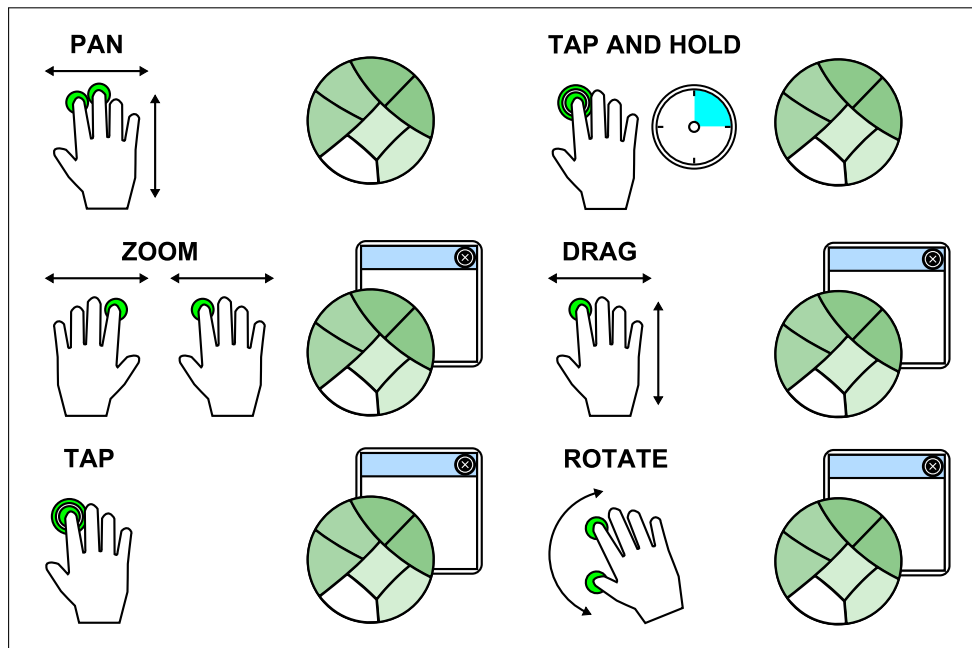


Abbildung 6.14: Interaktionstechniken mit Multi-Touch-Gesten

Zoom wird ebenso durch die Berührung mit zwei Fingern ausgeführt und skaliert bezüglich des Berührungsbereichs das gesamte Layout oder die Informationsfenster. Der Skalierungsfaktor wird entsprechend der Distanz zwischen den Eingabepunkten modifiziert.

Tap ist die kurzzeitige Berührung des Touchscreen mit einem Finger. Mit einem Tap können Layoutkomponenten aktiviert und selektiert sowie Schaltflächen betätigt werden.

Tap and Hold wird durch die Berührung mit einem Finger und gleichbleibender Eingabeposition über eine festgelegte Zeitspanne ausgelöst. Nach Ablauf der Zeitspanne und erfolgreicher Gestenvollendung können elementbezogene Informationsfenster aufgerufen werden.

Drag beschreibt die fortwährende Berührung mit einem Finger und gleichzeitige Bewegung des Eingabepunktes. Drag wird für die Aktivierung von Layoutkomponenten und die Verschiebung der Informationsfenster verwendet.

Rotate wird durch die Berührung mit zwei Fingern und die Bewegung um den Mittelpunkt zwischen den Eingabepunkten ausgeführt. Mit der Geste werden Drehungen des Layouts und der Informationsfenster um den Mittelpunkt der Eingabe realisiert.

Der implementierte Funktionsumfang umfasst vorrangig **Online-Gesten** und beinhaltet somit ausschließlich direkte Manipulationen. Durch Realisierung weiterer Eingabeprozessoren können hierzu Offline-Gesten ergänzt werden. Offline-Gesten werden typischerweise erst nach Vollendung der Interaktion verarbeitet und haben kontextbezogene Bedeutung. Auf diese Weise können durch Zeichnen von Freiformen und Strichfolgen komplexe Selektionen durchgeführt oder Systemfunktionen aufgerufen werden. Die definierten Gesten und verbundenen Aktionen des VIS sind kombinierbar und ermögli-

chen simultane Ausführungen. Dies erzeugt gegenüber Standardeingabegeräten meist eine schnellere und effizientere Interaktion mit den dargestellten Informationen. Die unabhängige Bedienung von Layout, Fenstern und GUI-Elementen fördert ein kollaboratives Arbeiten mit dem VIS. Mehrere Anwender können so gleichzeitig und gemeinsam interaktive Analyseprozesse ausführen und die visuellen Informationen individuell anpassen. Um den Benutzern in diesem Zusammenhang eine natürliche Bedienung zu ermöglichen, besitzen die Bedienbestandteile eine simulierte Bewegungsträgheit. Führt der Nutzer eine schnelle Verschiebeaktion durch, werden die Komponenten eine kurze Distanz mit abnehmender Geschwindigkeit in die Aktionsrichtung weiterbewegt. Dies erweckt den Eindruck der Interaktion mit realen Objekten und fördert das Erleben des Informationsgehalts.

7 Evaluation und Validierung

Die Ergebnispräsentation und das dazu verwendete Visualisierungsverfahren des VIS sind primär, wie in Kapitel 4 definiert, von Zielgruppe und Einsatzfeld abhängig und müssen sich an den grundsätzlichen Anforderungen bezüglich Expressivität, Effektivität und Angemessenheit orientieren. Insbesondere die Berücksichtigung verschiedener Zielgruppen bedeutet häufig, dass ein darzustellender Sachverhalt in unterschiedlichen Detaillierungsstufen und Darstellungsformen aufbereitet werden muss oder möglichst allgemeinverständlich und begreifbar zu visualisieren ist. Ein flexibles System schafft daher idealerweise eine Kommunikationsbasis für Benutzer mit unterschiedlichen Fachkenntnissen und ist für alle Beteiligten grundlegend erfassbar. Verschiedene Einsatzfelder und Zielgruppen können so beispielsweise von fachlichen Kennzahlen, wie auch von umfassendem Gesamtüberblick profitieren.

In diesem Zusammenhang wird eine **explorative Benutzerstudie** mit dem entwickelten System durchgeführt. Dies ermöglicht primär die Konkretisierung grundlegender Merkmale und Wirkhypothesen und die Identifizierung und Operationalisierung relevanter Variablen. Neben der allgemeinen Eruiierung der Gebrauchstauglichkeit und Interaktionsfunktionalität dient die Benutzerstudie schwerpunktmäßig der Untersuchung von Eignung und Eigenschaften des VIS. Insbesondere der Vergleich der entwickelten Knoten-Kanten- und Kapselung-Visualisierungskonzepte steht hierbei im Fokus. Bei der Ergebnisgewinnung, -analyse und -bewertung wird Bezug auf die spezifischen Anforderungen der Produktentwickler und Voraussetzungen der Produktstruktur genommen. Auf diese Weise werden Aussagen zur Angemessenheit des VIS ermöglicht.

Mit dieser Zielstellung wird eine Gruppe von zwölf freiwilligen Testpersonen ausgewählt. Zu den Probanden zählen repräsentative Fach- und Führungskräfte aus dem Bereich der Produktentwicklung oder Personen mit Bezug zur Bedienung von interaktiven Systemen und Kenntnissen über hierarchische Ordnungsstrukturen. Diese, aus den Branchen des Maschinenbaus, der Automobilindustrie, aus Forschungsinstituten und weiteren Bereichen stammenden Probanden, werden einem festgelegten Testverfahren unterzogen. Während des Testprozesses werden unter Verwendung von Datengewinnungsmethoden, wie Beobachtungen, Experimente, Interviews und quantitativer Methoden, objektive und subjektive Testdaten aufgenommen. Die objektiven Daten dienen der messbaren Betrachtung der Visualisierungs- und Interaktionskonzepte. Die persönlichen Beurteilungen, Vorurteile und Gefühle der Probanden werden hierbei jedoch nicht berücksichtigt. Die subjektiven Daten repräsentieren hingegen die individuelle Einstellung der Probanden, deren Vorwissen, Meinung und Fähigkeiten. Das Testverfahren und die statistische und qualitative Auswertung der aufgenommenen Testdaten wird in den folgenden Abschnitten beschrieben.

7.1 Hypothesen zur Benutzerstudie

Im Vorfeld der Benutzerstudie werden zunächst einige generelle Hypothesen bezüglich der entwickelten Visualisierungslayouts definiert. Diese Thesen basieren auf den in Abschnitt 2.6 in den Tabellen 2.3 und 2.4 beschriebenen grundlegenden Eigenschaften und Merkmalen der Visualisierungskonzepte und werden mit Beobachtungen und Erkenntnissen aus der Entwicklungsphase ergänzt. Die Thesen stehen ferner in einem sich gegenseitig bedingendem Zusammenhang und sind auf eine allgemeine Betrachtung der Wirkzusammenhänge beschränkt. Die folgenden acht exemplarischen Annahmen werden im Kontext der Benutzerstudie getroffen und ermöglichen eine hypothetische Betrachtung des VIS:

- H 1** Die Qualität der Aufgabenbewältigung und die benötigten Ressourcen besitzen eine Abhängigkeit zu den verwendeten Interaktionstechniken und zum verwendeten Präsentationsmodus der Layouts.
- H 2** Vorkenntnisse zu klassischen Konzepten von Baumstrukturen und die Verbreitung von herkömmlichen Knoten-Kanten-Layouts resultieren im Vorteil des Knoten-Kanten-Konzepts. In diesem Zusammenhang erzielt das Knoten-Kanten-Layout allgemein eine höhere Ergebnisqualität.
- H 3** Die Kenntnis über das Kapselung-Konzept ermöglicht eine effizientere Ermittlung und Identifikation von Layoutkomponenten. Dies resultiert aus der Effektivität und Expressivität der Komponentenkodierung, aus der Raumausnutzung und der differenzierbaren Komponentenrepräsentation des Layouts.
- H 4** Das Kapselung-Konzept ermöglicht effiziente Abschätzungs- und Vergleichsoperationen von speziell kodierten Komponentenparametern, wie die kummulative Anzahl von untergeordneten Komponenten. Das Knoten-Kanten-Konzept ist hingegen für das direkte Abzählen von verbundenen Komponenten geeignet.
- H 5** Durch die räumliche Konfiguration und dem resultierenden Leerraum erfordert das Knoten-Kanten-Konzept eine höhere Anzahl an Layoutinteraktionen, wie Panning und Zoom Aktionen.
- H 6** Das Knoten-Kanten-Konzept ist durch die direkte Abbildung der Strukturtopologie für die Analyse von bestimmten strukturellen Zusammenhängen besonders geeignet. Die explizite Darstellung von Gruppenkomponenten ist im Kapselung-Konzept auf den Explorationsmodus beschränkt und in diesem Zusammenhang von Nachteil.
- H 7** Im Kontext des Kapselung-Konzepts ist der Explorationsmodus für die Interaktion mit Gruppenkomponenten und für weitere spezifische strukturbezogene Aufgaben erforderlich und wird folglich verstärkt genutzt. In Bezug auf die vordefinierten Strukturlayouts der Benutzerstudie wird im Knoten-Kanten-Konzept der Übersichtsmodus hingegen verstärkt verwendet.
- H 8** Die Ermittlung von Ausgangskomponenten kann die Aussagekraft spezifischer Ergebnisdaten eventuell beeinträchtigen. Analog zu Hypothese H 1 erzielen die Probanden mit höherem Interaktionsverständnis gleichzeitig effizientere Ergebnisdaten.

7.2 Testverfahren und -ablauf

Der Hauptschwerpunkt der empirischen Benutzerstudie ist der Vergleich und die Untersuchung der entwickelten Visualisierungskonzepte. Die Probanden bearbeiten im Testverfahren daher je ein entwickeltes Visualisierungskonzept, wodurch zwei zufällig zusammengestellte Testgruppen für Knoten-Kanten- und Kapselung-Layout entstehen. Um vergleichbare Bedingungen zu schaffen, ist der Testaufbau und -ablauf, mit Ausnahme des Visualisierungstyps, einheitlich gestaltet. In diesem Zusammenhang werden der Funktionsumfang und bestimmte Bestandteile des VIS erweitert und angepasst und auf das Testverfahren abgestimmt. Nicht relevante Funktionen und Bestandteile werden hingegen deaktiviert, um eine zielgerichtete Benutzerführung zu realisieren und den Fokus der Probanden auf die Aufgabenbearbeitung und dargestellten Informationen zu richten. Zur Vermeidung etwaiger Fehlbedienungen wurden beispielsweise bestimmte Bestandteile, wie die Layoutparametrisierung und -generierung, die Elementpräsentations- und Farbkodierungsvarianten und die Systemeinstellungen als konstant definiert. Weiterhin beschränken sich die Interaktionsmethoden auf ein Standardeingabegerät, wobei die erweiterten Multi-Touch Funktionen den Probanden nicht zur Verfügung stehen. Alle Aktionen im Testverfahren werden von den Probanden mittels einer Maus als Eingabegerät gesteuert und ausgeführt. Durch den gewohnten Umgang mit der Maus und die Kenntnis über interaktive Systembedienung, können die Probanden nach dem Erlernen des testbezogenen Aktionsschemas intuitiv mit den Prototypen interagieren.

Der Testablauf wird von einem **Testbetreuer** beaufsichtigt und geleitet. Dieser ist für die Unterstützung und Erklärung des Testverfahrens, der Aufgabenstellungen, zusätzlicher Randbedingungen und für die Steuerung der Datenaufnahme verantwortlich. Während der Aufgabenbearbeitung erhebt der Testbetreuer die freiwillig abgegebenen, objektiven und subjektiven Daten des Probanden und fördert weiterführende Aussagen, um vollständige Meinungen und Einschätzungen zu erhalten. Der Betreuer nutzt neben Beobachtung und Befragung der Probanden vordergründig zwei Techniken zur Datenerhebung, welche folgend beschrieben werden:

Der Fragebogen ist ein breit eingesetztes Instrument der Datenerhebung. In der Benutzerstudie wird eine Mischung aus teilstandardisiertem und standardisiertem Fragebogen verwendet und für alle beteiligten Probanden einheitlich gestaltet. Die Befragten beantworten vorrangig offene Fragen mit freier Antwortwahl und geben skalierte Einschätzungen zu Ergebnissen und Aussagen anhand von Bewertungsskalen. Der Fragebogen dient weiterhin zur Protokollierung von bestimmten Ergebnisdaten des Testverfahrens und fungiert folglich zur Erfassung von subjektiven und objektiven Daten. Anhang A enthält eine Abschrift eines beispielhaften Fragebogens mit exemplarischen Ergebnisdaten aus der Benutzerstudie.

Die Logdatei ist ein typischerweise automatisch geführtes Ereignisprotokoll aller oder bestimmter Aktionen von Prozessen auf einem Computersystem. Im Kontext des Testverfahrens wird für jeden Probanden eine individuelle Logdatei erzeugt, worin

alle relevanten Bedienaktionen, -zeiten und weiteren Werte des VIS im Verlauf der prototypbezogenen Aufgabenbearbeitung protokolliert werden. Das resultierende Protokoll umfasst den Hauptanteil der objektiven Daten und wird in der folgenden Auswertung verarbeitet. In Tabelle 7.1 ist der prinzipielle Aufbau der Logdatei auszugsweise beschrieben.

```

1  <evaluation type="enclosure" id="enclosure_eval_2010_11_10_18_24">
2    ...
3    <task number="3" timeTotal="46596">
4      <time>
5        <total><ms>46596</ms></total>
6        <browserMode><ms>0</ms><norm>0.0</norm></browserMode>
7        <iactMode><ms>46596</ms><norm>1.0</norm></iactMode>
8        <graphDrag><ms>36</ms><norm>7.725985E-4</norm></graphDrag>
9        <infoPanelDrag><ms>0</ms><norm>0.0</norm></infoPanelDrag>
10     </time>
11     <interaction>
12       <total><count>12</count></total>
13       <graphZoom><count>0</count><norm>0.0</norm></graphZoom>
14       <graphDrag><count>1</count><norm>0.083333336</norm></graphDrag>
15       <nodeSelect><count>5</count><norm>0.41666666</norm></nodeSelect>
16       <infoPanelOpen><count>6</count><norm>0.5</norm></infoPanelOpen>
17       <infoPanelDrag><count>0</count><norm>0.0</norm></infoPanelDrag>
18     </interaction>
19     <gui>
20       <total><count>12</count></total>
21       <resetBtn><count>0</count><norm>0.0</norm></resetBtn>
22       <iactModeBtn><count>3</count><norm>0.25</norm></iactModeBtn>
23       <selectBackBtn><count>9</count><norm>0.75</norm></selectBackBtn>
24     </gui>
25   </task>
26   ...
27 </evaluation>

```

Tabelle 7.1: Ausschnitt einer Logdatei der Benutzerstudie im XML-Format

Der Aufbau der Benutzerstudie richtet sich nach einem festgelegten Schema und orientiert sich am Aufbau des Fragebogens. Im Ablauf werden folglich zunächst testbezogene Identifikationsdaten, wie Zeitpunkt, laufende Teilnehmerkennung (Teilnehmer-ID) und Visualisierungstyp aufgenommen. Diese dienen der nachfolgenden Zuordnung und Kategorisierung der Evaluationsergebnisse. Ferner werden allgemeine demografische Daten der Probanden festgehalten und Angaben über Vorkenntnisse im Bereich der Produktentwicklung und im Umgang mit Strukturen ergänzt. Die Datenerhebung ist grundsätzlich anonym und die Probanden werden über den Datenschutz informiert. Der Testbetreuer erläutert daraufhin den generellen Ablauf der Aufgabenbearbeitung, die zugelassenen Hilfestellungen und Materialien. Die Probanden werden aufgefordert, die Aufgaben möglichst zügig zu bearbeiten, Verständnisfragen im Vorfeld zu klären und die Aufgaben-

bearbeitung bei fehlerhafter Beantwortung fortzuführen. Der anschließende Testverlauf gliedert sich in die Einführung in das VIS, einleitende Strukturtests, Aufgabenbearbeitung mit dem VIS und eine abschließende Einschätzung und Bewertung.

Die **Einführung in das VIS** umfasst die Erläuterung der Interaktionsmethoden, Bedienelemente und Merkmale des Layouts. Der Testbetreuer erklärt dabei primär die in Abschnitt 6.4 vorgestellten Bestandteile und demonstriert den Umgang mit dem VIS. Die strukturellen Bedeutungen und Prinzipien der dargestellten Informationen und die verschiedenen Ausgangssituationen und Voraussetzungen in der Benutzerstudie werden beschrieben. In einer festgelegten Zeitspanne hat der Proband die Möglichkeit, sich eigenhändig mit der Bedienung des Prototypen vertraut zu machen und gegebenenfalls auftretende Unklarheiten zu klären. Die Einführung in das VIS wird mittels vorgefertigter Layouts identischer Strukturen durchgeführt, welche in Abbildung 7.1 dargestellt sind.

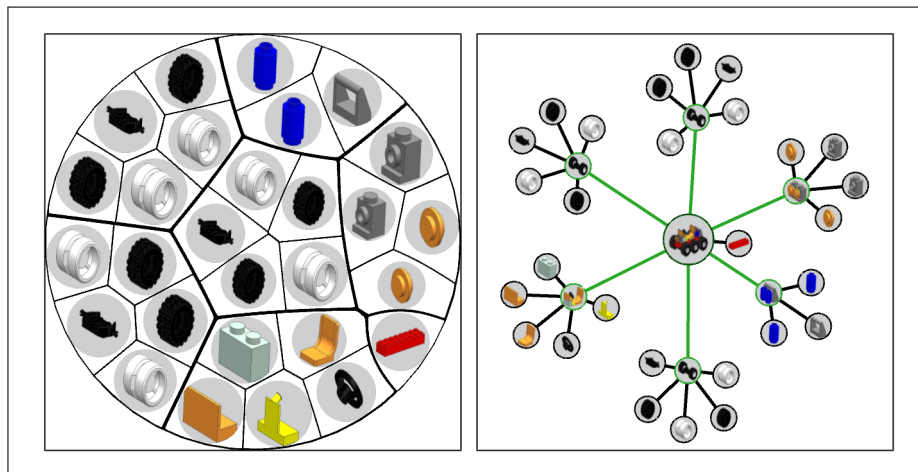


Abbildung 7.1: Einführende Visualisierungslayouts der Benutzerstudie. Die Layouts zeigen die Produktstruktur eines Legoautos bestehend aus 28 Bauteilen, 7 Baugruppen und 3 Hierarchieebenen.

Im Anschluss an die Einführung in das VIS folgen zunächst **einleitende Strukturtests**, um die verwendeten Terminologien und Aufgabenstellungen zu konkretisieren. Der 1. Strukturtest umfasst exemplarische Aufgabentypen und Begrifflichkeiten, welche mittels klassischer **Indent-Layouts** vorgestellt werden. Zu Beginn des 1. Strukturtests erklärt der Testbetreuer die Aufgabenstellung anhand eines Beispiellayouts und fünf vordefinierter Aussagen. Die 1. Aufgabenstellung beinhaltet fünf Fragen zu je unterschiedlichen Indent-Layouts und ist von den Probanden zu bearbeiten. In diesem Zusammenhang werden jedem Strukturelement bzw. Layout-Knoten zufällige Nummerierungen zugeordnet, welche als Antwortdaten dienen. Der Testbetreuer organisiert den Ablauf der Aufgabenbearbeitung und misst die benötigte Zeit bis zur Aufgabendovollendung. Die verwendeten Indent-Layouts sind auszugsweise in Abbildung 7.2 dargestellt.

Im 2. Strukturtest wird der Proband aufgefordert, eine Layoutdarstellung bezüglich des Visualisierungstyps in ein äquivalentes Indent-Layout umzuwandeln. Die Kapselung-

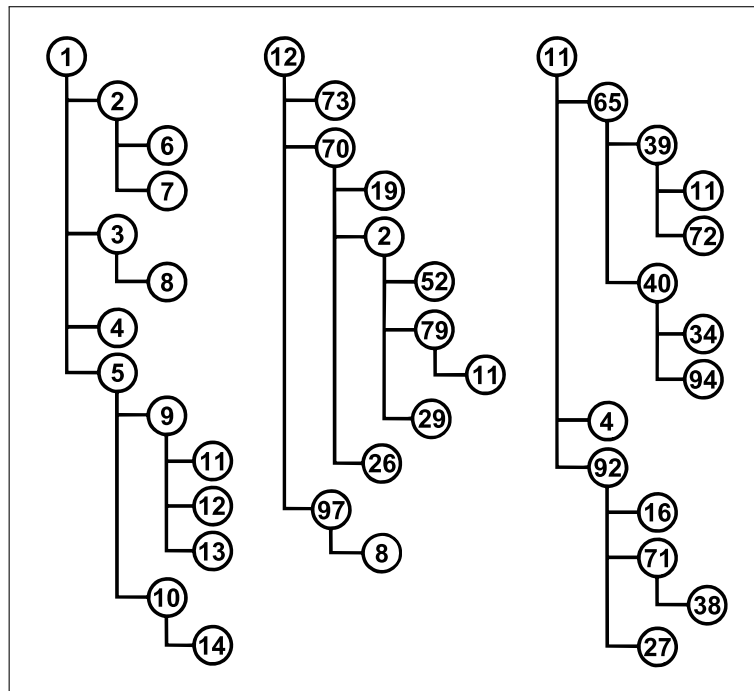


Abbildung 7.2: Erster einleitender Strukturtest der Benutzerstudie. Abgebildet sind beispielhafte Indent-Layouts der 1. Aufgabenstellung.

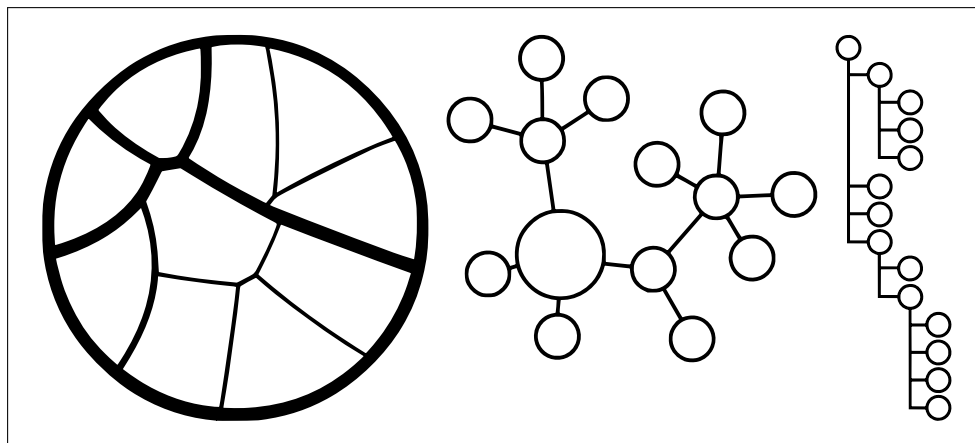


Abbildung 7.3: Zweiter einleitender Strukturtests der Benutzerstudie. Abgebildet sind die entsprechend des Visualisierungstyps vorgelegten Layoutformen der 2. Aufgabenstellung (links und mitte) und ein mögliches Ergebnis nach der Transformation in das Indent-Layout (rechts).

und Knoten-Kanten-Layouts werden hierbei im Übersichtsmodus dargestellt und sind von den Probanden unter dieser Voraussetzung zu bearbeiten. Die strukturelle Korrektheit des übertragenen Layouts wird überprüft und die benötigte Bearbeitungszeit der 2. Aufgabenstellung wird aufgenommen. Abbildung 7.3 zeigt die hierzu verwendeten Layoutformen.

Die nächste Phase des Testverfahrens behandelt die **Aufgabenbearbeitung mit dem VIS**. Der Testbetreuer erläutert die allgemeine Ausgangssituation für die 3. Aufgabenstellung und erklärt den Testablauf mit dem Prototyp. Vor dem Bearbeitungsbeginn jeder Aufgabe macht sich der Proband zunächst mit der Aufgabenstellung vertraut und erhält das benötigte Aufgabenmaterial. Das Aufgabenmaterial besteht generell aus schriftlichen Informationen, wie Name, Komponentennummer, eindeutiger Identifikationsnummer zu Bauteilen oder -gruppen und den zugehörigen bildlichen Darstellungen der Strukturelemente. Die gesamte Aufgabenstellung bezieht sich auf eine vordefinierte Produktstruktur und die entsprechende Layoutdarstellung, wobei jede Frage unterschiedliche Strukturbestandteile und Layoutaspekte betrifft. Das Aufgabenschema orientiert sich ferner an möglichen Anwendungsszenarien in der Produktentwicklung und den allgemeinen Handlungs- und Arbeitsprozessen mit Produktstrukturen. Dies umfasst beispielsweise Ermittlung, Identifikation, Vergleiche und Analyse von Bauteilen und -gruppen. In der Regel sind darüber hinaus die zugehörigen Attribute, Kontexte und strukturellen Zusammenhänge von Interesse. Die verwendeten Layoutdarstellungen sind in Abbildung 7.4 illustriert. Der zugehörige Prototyp wird vor und zwischen jeder Aufgabenbearbeitung in einen Leerlaufmodus versetzt, in dem keine Informationen dargestellt werden und die Bedienung deaktiviert ist. Gleichzeitig wird der Ausgangszustand des Visualisierungslayouts hergestellt, der Explorationsmodus aktiviert und alle Komponenten werden deaktiviert und deselektiert. Auf diese Weise wird der Proband veranlasst sich erneut zu orientieren und die Informationssuche mittels der verfügbaren Navigationsmethoden und Bedienelemente durchzuführen. Sobald der Proband für die Bearbeitung bereit ist, wird das VIS durch den Testbetreuer aktiviert und die interne Protokollierung der objektiven Daten, aller relevanten Aktionen des Probanden beginnt. Nachdem die Aufgabe korrekt gelöst, das Ergebnis notiert und die Vollendung bekannt gegeben wurde, kehrt der Prototyp in den Leerlaufmodus und in die Ausgangssituation zurück. Nimmt der Proband hingegen ein falsches Ergebnis, wird die Aufgabenbearbeitung nach einem Hinweis durch den Testbetreuer fortgeführt und der Fehlversuch notiert. Im Anschluss an jede Aufgabenvollendung wird der Proband aufgefordert, den Schwierigkeitsgrad der Aufgabe anhand einer Bewertungsskala einzuschätzen. Die 3. Aufgabenstellung umfasst sieben Aufgaben, welche folgend beschrieben sind:

A 3.1 Eine festgelegte atomare Komponente ist anhand gegebener Eigenschaftsdaten und des Komponentenikons eindeutig zu ermitteln.

A 3.2 Als besondere Ausgangssituation ist eine Komponente vorselektiert, vergrößert und zentriert. Zu dieser Komponente sind alle übergeordneten Komponenten bis

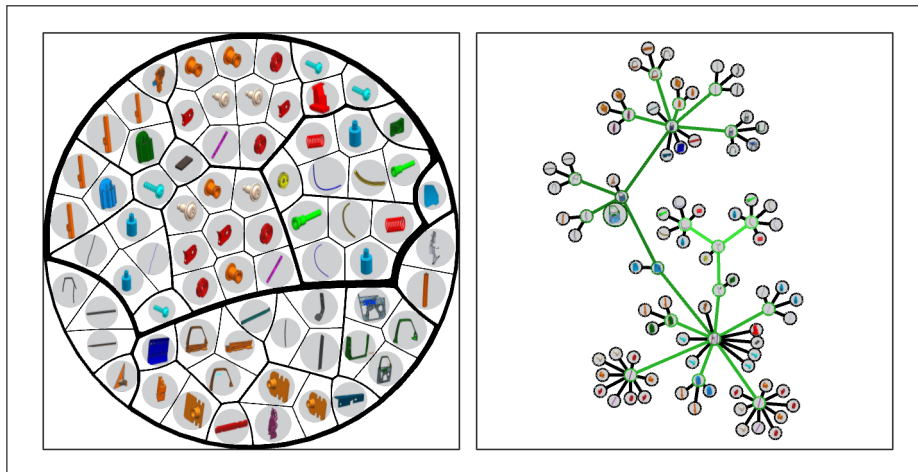


Abbildung 7.4: Visualisierungslayouts der Benutzerstudie für die Aufgabenbearbeitung mit dem VIS. Die Layouts zeigen die Produktstruktur einer Autotür bestehend aus 73 Bauteilen, 21 Baugruppen und 8 Hierarchieebenen.

zur Wurzelkomponente zu bestimmen. Die farbliche Hervorhebung des Navigationspfads ist bei dieser Aufgabe deaktiviert.

- A 3.3** Die Menge der direkten und indirekten Subkomponenten von zwei vordefinierten Ausgangskomponenten ist zu vergleichen. Gefordert ist die Identifizierung der Ausgangskomponente mit den meisten Subkomponenten jedoch nicht zwingend deren Anzahl. Als Hilfestellung zur Suche nach den Ausgangskomponenten sind im Aufgabenmaterial deren Elternkomponenten namentlich angegeben.
- A 3.4** Mit den gleichen Voraussetzungen ist analog zu Aufgabe A 3.3 die Komponente mit den meisten Subkomponenten zu bestimmen. Zusätzlich ist die genaue Anzahl der Subkomponenten gefordert.
- A 3.5** Die Anzahl einer Komponente in der Gesamtstruktur ist zu bestimmen. Für die Differenzierung dienen der Name, bildliche Darstellung und Nummer der Komponente. Die eindeutigen Identifikationsnummern sind jedoch zu vernachlässigen.
- A 3.6** Die Anzahl der direkten Subkomponenten zu einer festgelegten Komponente ist zu bestimmen. Als Hilfestellung sind Elternkomponenten der Ausgangskomponente namentlich angegeben.
- A 3.7** Zwei strukturell geringfügig verschiedene Layoutdarstellungen einer Ausgangsstruktur sind zu vergleichen. Die räumliche Konfiguration der Layoutkomponenten ist dabei zu vernachlässigen und ausschließlich die strukturelle Differenz ist als Ergebnis zu notieren. In Zusammenhang mit dieser Aufgabe werden für die Bearbeitung zwei unabhängig zu bedienende Prototypen zur Verfügung gestellt.

Im abschließenden Teil der Benutzerstudie werden **Einschätzungen und Bewertungen** über den allgemeinen Evaluationsprozess und das präsentierte VIS erhoben. In diesem Zusammenhang werden die Probanden aufgefordert, persönliche Stellungnahmen zum Umgang mit dem VIS und zu Impressionen und Beobachtungen während der Aufgabenbearbeitung anzugeben. Zunächst werden die Probanden mit vordefinierten

Aussagen konfrontiert, welche bezüglich der eigenen Einschätzung zu bewerten sind. Die Beurteilung wird mittels Bewertungsskalen vorgenommen und umfasst die folgenden 21 Aussagen:

- E 1** Die Layoutdarstellung der Struktur ist verständlich und mit minimaler Einarbeitungszeit nachvollziehbar.
- E 2** Die strukturellen Beziehungen in der Layoutdarstellung sind intuitiv verständlich und nachvollziehbar.
- E 3** Im Verlauf der Aufgabenbearbeitung entsteht ein Lern- bzw. Erinnerungsprozess zur abgebildeten Struktur.
- E 4** Falls E 3 zutreffend – Die Strukturkenntnis ist im weiteren Verlauf der Aufgabenbearbeitung hilfreich und nachfolgende Aufgaben sind leichter zu lösen.
- E 5** Für die Aufgabenbearbeitung sind keine weiteren Hilfestellungen nötig.
- E 6** Die räumliche Konfiguration der Layoutdarstellung und die Komponentenpositionen sind verständlich.
- E 7** Die Layoutdarstellung ermöglicht eine intuitive Strukturnavigation und einfache Orientierung selbst bei starker Vergrößerung des Layouts.
- E 8** Die vorgestellte Layoutdarstellung wird gegenüber einem Indent-Layout bevorzugt.
- E 9** Die Strukturdarstellung und die Farbkodierung sind ästhetisch ansprechend.
- E 10** Die farbliche Kodierung ist für die Orientierung in der Struktur hilfreich und unterstützend.
- E 11** Die bildliche Komponentendarstellung ist für die Identifikation und das Strukturverständnis hilfreich und unterstützend.
- E 12** Die Größe der Zellen (Kapselung) bzw. Länge der Kanten (Knoten-Kanten) ist für Aufgabenbearbeitung und Strukturverständnis hilfreich und unterstützend.
- E 13** Der Übersichtsmodus ist hilfreich und unterstützend.
- E 14** Der Explorationsmodus ist hilfreich und unterstützend.
- E 15** Die verfügbaren Interaktionsmethoden, wie Zoom, Panning, Autofokus, etc., sind ausreichend.
- E 16** Die halbtransparente Vorschau der Subkomponenten bei der Aktivierung einer Gruppenkomponente ist hilfreich und für die Interaktion unterstützend.
- E 17** Der Autofokus und -zoom bei der Selektion von Komponenten ist hilfreich und für Interaktion und Orientierung unterstützend.
- E 18** Die Informationsfenster sind für die Darstellung der Informationen von Komponenten gut geeignet.
- E 19** Die Informationsfenster sind intuitiv zu bedienen.
- E 20** Nur Kapselung: Die auf den Explorationsmodus beschränkte Darstellung und Selektion von Gruppenkomponenten ist nicht von Nachteil.
- E 21** Nur Kapselung: Die adaptiven Randlinien der Zellen sind für das Strukturverständnis und die Identifikation von Geschwisterkomponenten hilfreich.

Im Anschluss an die Bewertung der Aussagen ist es den Probanden möglich, weitere freie Einschätzung und Stellungnahmen abzugeben. In der Konversation mit dem Testbetreuer werden hierbei weiterführende Betrachtungen und Optimierungsaspekte diskutiert und protokolliert. Die entstehenden subjektiven Daten umfassen verschiedene Aspekte des VIS, wie die Eignung und Qualität von Bedienung und dargestellten Informationen. Diese Daten dienen der Bemessung der allgemeinen Gebrauchstauglichkeit und der in Abschnitt 4.1 angeführten Kriterien von Expressivität, Effektivität und Angemessenheit.

7.3 Auswertung der Ergebnisdaten

Die Benutzerstudie wurde von den beteiligten zwölf Probanden erfolgreich durchgeführt, und es entstanden je sechs Ergebnisdatensätze für das Kapselung- und Knoten-Kanten-Konzept. Im Rahmen des definierten Testverfahrens und -ablaufs wurden dabei objektive und subjektive Daten der Probanden erhoben, welche umfassende Auswertungsansätze zu verschiedensten Aspekten des VIS und dessen Bestandteilen bieten. Die verfolgte Zielstellung der Evaluation ist die Validierung in Bezug auf die Softwarequalitätssicherung anhand der aufgenommenen Daten und die Berechnung von Eignungskriterien der entwickelten Software. Die im Kontext der Arbeit definierten Bestandteile der Benutzerstudie umfassen eine Auswahl von möglichen Analyseaspekten und bestimmten Betrachtungskriterien. Die Ergebnisdaten sind zudem partiell durch die festgelegten Randbedingungen der Aufgabenstellungen und Datenerhebung bedingt und unter Berücksichtigung dieser Aspekte zu betrachten. Die vorgenommene Auswertung der gewonnenen Ergebnisse richtet sich somit nach der generellen Validierung und der Erzeugung von Tendenzen, Richtlinien und Faktoren für die weiterführende Entwicklung des VIS. In diesem Abschnitt werden folgend die wesentlichen Beobachtungen der Benutzerstudie und eine Auswahl der erzielten Ergebnisdaten präsentiert. Um eine allgemeine Betrachtung zu ermöglichen und generelle Tendenzen zu analysieren, werden hierbei primär gewichtete Mittelwerte bezüglich der Gesamtheit der Probanden bzw. der beiden Probandengruppen ermittelt. Eine weiterführende und probandenspezifische Analyse der Ergebnisdaten steht für anschließende Entwicklungen, Forschungsarbeiten und Fortführung der Benutzerstudien zur Verfügung.

Die **Auswertung des Fragebogens** umfasst zunächst die Analyse der Probandenkenn-daten. In diesem Zusammenhang wurden insbesondere Vorkenntnisdaten der Probanden bezüglich dem Umgang mit hierarchischen Strukturen, mit virtuellen Baugruppen und Produktstrukturen und zu Erfahrungen im CAD Umfeld erhoben. In Abbildung 7.5 sind die berechneten Probandendaten dargestellt. Die Analyse der Daten zeigt, dass die Probanden grundsätzlich einer möglichen Zielgruppe zuzuordnen sind und somit eine repräsentative Betrachtung des VIS durchgeführt werden kann. Bei zwei Probanden beschränken sich die Vorkenntnisse auf den Umgang mit hierarchischen Strukturen. Die Mehrheit der Probanden kann hingegen, sowohl Erfahrungen im CAD Umfeld, als auch mit virtuellen Baugruppen vorweisen.

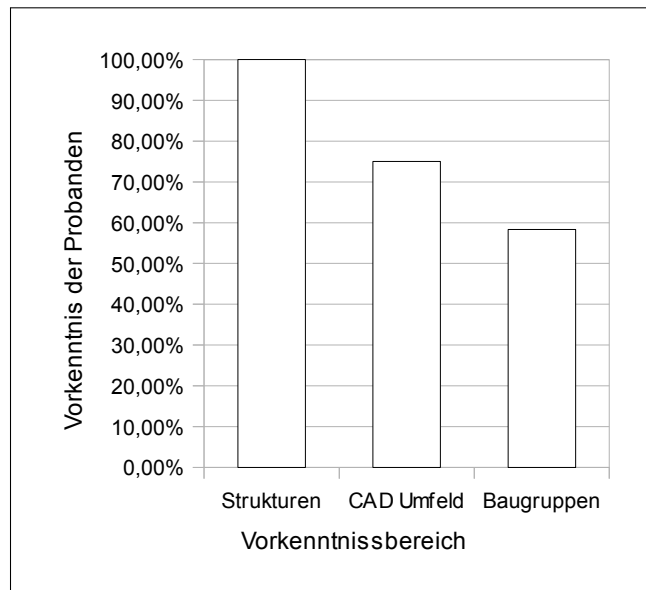


Abbildung 7.5: Prozentuale Vorkenntnisse der Probanden in den Bereichen hierarchische Strukturen, CAD und virtuelle Baugruppen

Die nach der Einführung in das VIS durchgeführten **Strukturtests** wurden von allen Probanden mit keinen oder nur geringen Fehlern absolviert. Der 1. Strukturtest beinhaltete fünf strukturbezogene Aufgabenstellungen zu verschiedenen Indent-Layouts und war für die gesamte Benutzerstudie einheitlich gestaltet. Die Bearbeitungszeit dieses Tests lag im Mittel bei 58 Sekunden und weist keine signifikanten Abweichungen auf. Die geringe Fehlerrate und die konsistenten Bearbeitungszeiten unterstreichen die grundlegenden strukturbezogenen Vorkenntnisse der Probanden. Im 2. Strukturtest wurden die Probanden aufgefordert ein Kapselung- bzw. Knoten-Kanten-Layout in eine äquivalente Indent-Struktur zu transformieren. Die durchschnittlichen Bearbeitungszeiten bezüglich der Visualisierungskonzepte sind in Abbildung 7.6 dargestellt. Die dem Kapselung-Konzept zugeordneten Probanden hatten laut Aussagen weniger Vorkenntnisse über das Prinzip des Layoutkonzepts. Dennoch weisen die Zeiten nur minimale Abweichungen zueinander auf, wodurch auf ein annähernd gleichwertiges Verständnis über die Visualisierungskonzepte geschlossen werden kann. Durch die Bedingungen des Kapselung-Konzepts im Übersichtsmodus waren die Analyse und Bearbeitung ferner auf die entsprechende Präsentation der Layoutkomponenten beschränkt. Die Probanden orientierten sich folglich an der räumlichen Konfiguration und den adaptiven Randlinien zur Bestimmung der Gruppenkomponenten und der Gesamtstruktur. Diese besondere Randbedingung wirkte sich allerdings nicht nachteilig aus.

Die Auswertung der **Aufgabenbearbeitung mit dem VIS** wird anhand des Fragebogens und der Analyse des Ereignisprotokolls realisiert. Im Fragebogen wurden die Fehler und die Bemessungen des Schwierigkeitsgrades zu den jeweiligen Aufgaben protokolliert, und die Logdatei enthält Daten über ausgeführte Aktionen und Vorgehen der Probanden. Die Reihenfolge der zu bearbeitenden Aufgaben wurde im Testverfahren für jeden Probanden variiert, um gleichgewichtige Mittelwerte zu erhalten. Die Mehrheit

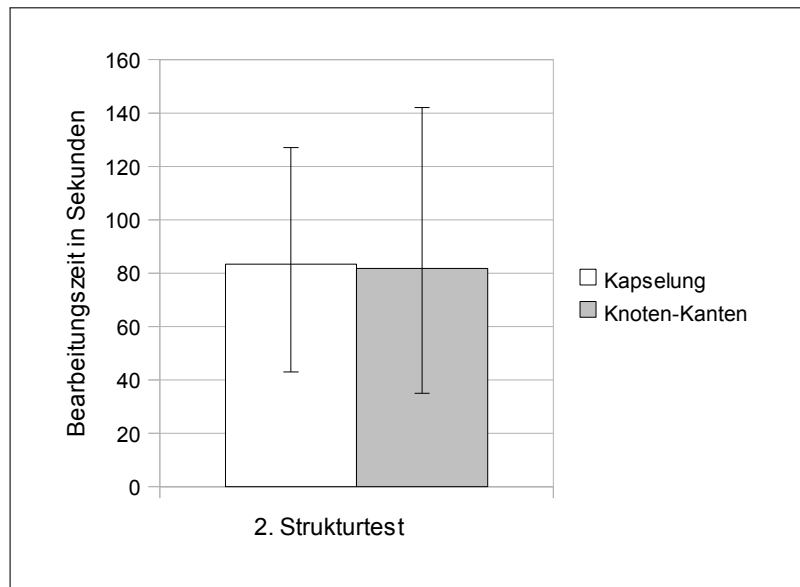


Abbildung 7.6: Durchschnittliche Bearbeitungszeiten und zugehörige extreme Abweichungen bezüglich der Visualisierungskonzepte im 2. Strukturtest

der Probanden bearbeitete die Aufgaben nicht korrekt und wurde durch den Testbetreuer auf Fehler hingewiesen. Zwei, jeweils dem Kapselung- und dem Knoten-Kanten-Konzept zugehörige Probanden, lösten die komplette Aufgabenstellung ohne Fehler. Beim Kapselung-Konzept liegt die **Fehlersumme** insgesamt bei neun Fehlern (21,43%) und beim Knoten-Kanten-Konzept traten insgesamt acht Fehler (19,05%) auf. Aus dem Beobachtungs- und Äußerungsprotokoll der Benutzerstudie geht hervor, dass die aufgetretenen Fehler partiell auf die Risikobereitschaft der Probanden zurückzuführen sind. Aufgrund der bekanntgegeben Zeitmessung während der Aufgabenbearbeitung neigten die Probanden teilweise dazu, erste Vermutungen als Ergebnis zu äußern, ohne diese zu überprüfen. Die geringen Abweichungen der Fehlersummen lassen darauf schließen, dass keines der beiden Visualisierungskonzepte eine grundsätzlich überwiegende Fehleranfälligkeit aufweist. Die durchschnittliche Fehlerrate der einzelnen Aufgaben ist in Abbildung 7.7 dargestellt und ermöglicht hingegen potenzielle Rückschlüsse auf die Eignung spezieller Visualisierungs- und Interaktionsbestandteile in Bezug zur Aufgabenstellung. Das Kapselung-Konzept weist vordergründig in den Aufgaben A 3.2 und A 3.3 eine erhöhte Fehlertendenz auf und das Knoten-Kanten-Konzept in der Aufgabe A 3.6. Anhand der protokollierten Beobachtungen während der Aufgabenbearbeitung lassen sich demnach beispielsweise folgende **Fehlerquellen** identifizieren:

F 1 – Unikal direkt verbundene Komponenten: Die Ausgangsstrukturen der Visualisierungslayouts besitzen gegebenenfalls die besondere Charakteristik, dass bestimmte Komponenten nur eine einzige direkt über- oder untergeordnete Komponente besitzen. Im umgesetzten Kapselung-Konzept besitzen diese unikal verbundenen Komponenten eine starke Ähnlichkeit in Form und Fläche zur über- bzw. untergeordneten Komponente. Im Kontext der Produktstrukturen repräsentieren die Layoutkomponenten Bauteile und aus Bauteilen zusammengesetzte

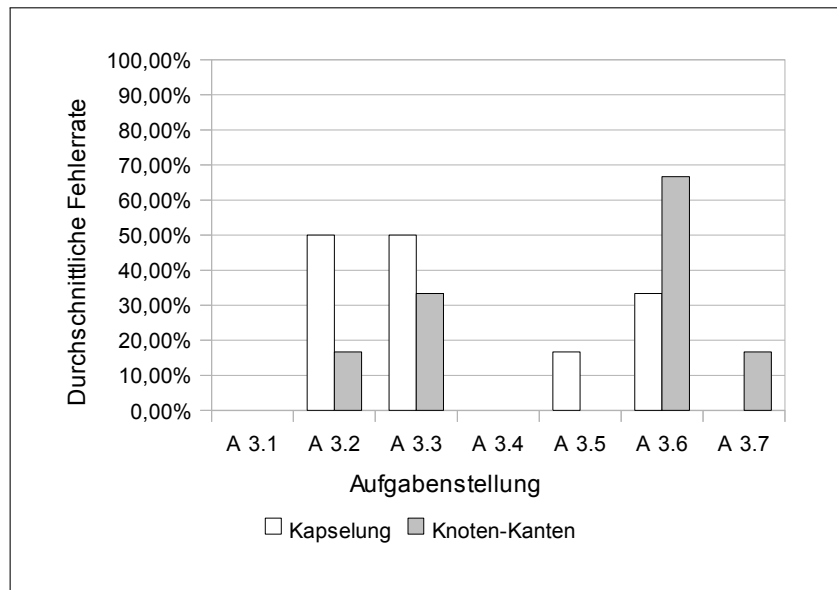


Abbildung 7.7: Durchschnittliche Fehlerraten der Aufgabenstellungen in der Aufgabenbearbeitung mit dem VIS

Baugruppen. Dieser Zusammenhang zwischen Bauteil und -gruppe führt weiterhin dazu, dass direkt über- oder untergeordnete Layoutkomponenten häufig vergleichbare Ikone besitzen, und die Identifikation und Differenzierung somit potenziell beeinträchtigt wird. In Aufgabe A 3.2 ist die Ermittlung aller Elternkomponenten zu einer vorausgewählten Komponente gefordert. Wie in Abbildung 7.8 dargestellt, trifft der beschriebene Fall auf die ersten beiden Komponenten in der verwendeten Ausgangsstruktur zu, wodurch die Probanden die Ermittlung bereits bei der untergeordneten Komponente abschlossen. In Aufgabe A 3.3 sind zwei Gruppenkomponenten zu vergleichen, um die Komponente mit den meisten Subkomponenten bis zum Verzweigungsende zu ermitteln. Wie aus Abbildung 7.8 ersichtlich, wurden die Probanden auch bei dieser Aufgabe durch ähnliche Form, Fläche und Ikone fehlgeleitet.

F 2 – Transparenz in der Navigations- und Verbindungsvorschau: Die Aktivierung einer Komponente im Explorationsmodus führt, wie in Abschnitt 6.4 beschrieben, zur halbtransparenten Verbindungs- bzw. Navigationsvorschau der direkt verbundenen Komponenten. Diese Funktion wurde zwar grundsätzlich als hilfreich eingestuft, konnte aber durch Beobachtungen und Äußerungen der Probanden auch als Fehlerquelle identifiziert werden. Die in der Benutzerstudie implementierte Funktionsversion wird im Kapselung-Konzept durch die Bewegung des Eingabepunktes in die Fläche einer Gruppenkomponente aktiviert. Die Farbkodierung und Ikone der Gruppenkomponente wird dabei durchsichtig und die untergeordneten Komponenten werden innerhalb der Komponentenfläche sichtbar. Abbildung 7.8 zeigt, dass diese Funktion unter Umständen zu Orientierungs- und Zuordnungs-

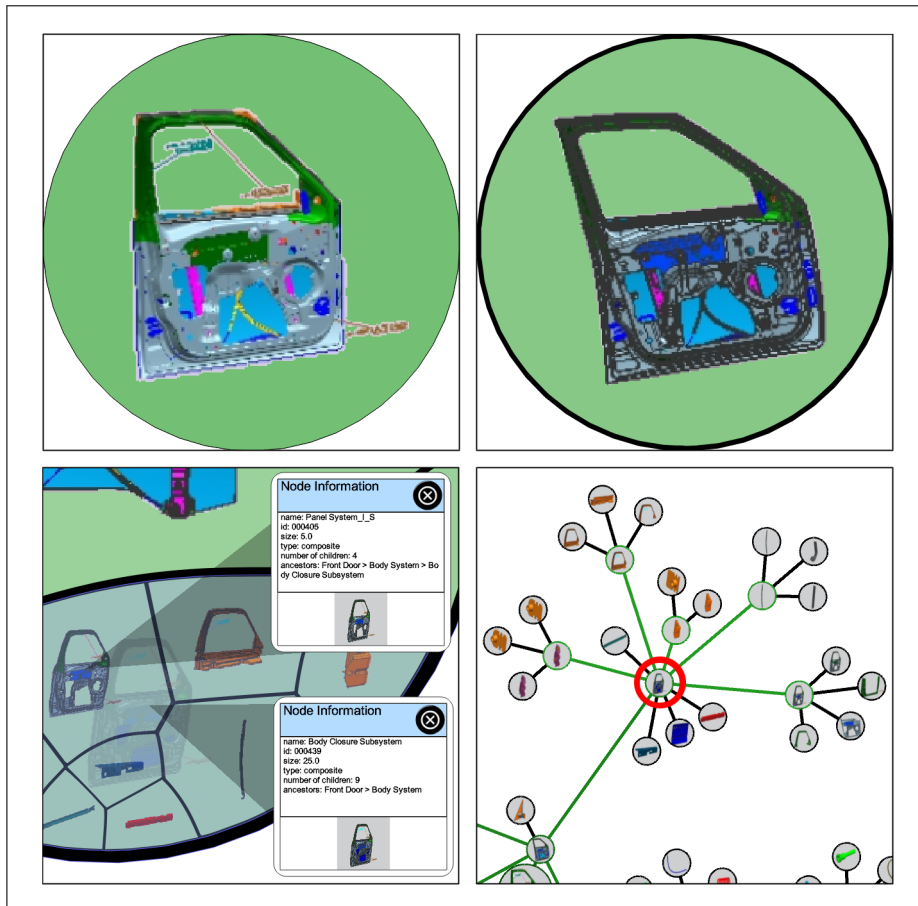


Abbildung 7.8: Potentielle Fehlerquellen in der Aufgabenbearbeitung mit dem VIS. Die oberen Abbildungen zeigen unikal direkt verbundene Komponenten mit Ähnlichkeit in Form, Fläche und Ikone. Die Abbildung unten links zeigt die Transparenz in der Navigations- und Verbindungsvorschau mit Beeinträchtigung der Orientierung sowie Zuordnungsprobleme zwischen Informationsfenster und Komponenten. Das obere Informationsfenster bezieht sich auf eine untergeordnete Komponente und das untere Fenster ist der halbtransparenten übergeordneten Komponente zugeordnet. Die Abbildung unten rechts zeigt eine Komponente (rot markiert) mit variablen Kantenlängen zu direkt verbundene Komponenten.

problemen führen kann. In diesem Zusammenhang wurden die Probanden teils bei der Bestimmung der Zugehörigkeit von Aktionen und Informationsfenstern zu Komponenten und Strukturebenen behindert.

F 3 – Variable Kantenlängen direkt verbundener Komponenten: Im Knoten-Kanten-Konzept besitzen bestimmte Gruppenkomponenten stark voneinander abweichenden Kantenlängen zu direkt verbundenen Komponenten. Den verbundenen Komponenten sind wiederum verschieden gewichtige Teilstrukturen untergeordnet, woraus im Generierungsprozess die unterschiedlichen Kantenlängen resultieren. Die über- bzw. untergeordneten Komponenten besitzen folglich eine große räumliche Distanz zueinander, wodurch beispielsweise Zuordnungs- bzw. Abzähloperationen erschwert werden. Abbildung 7.8 zeigt den beschriebenen Umstand, welcher sich in Aufgabe A 3.6 auf die Fehlerrate auswirkte.

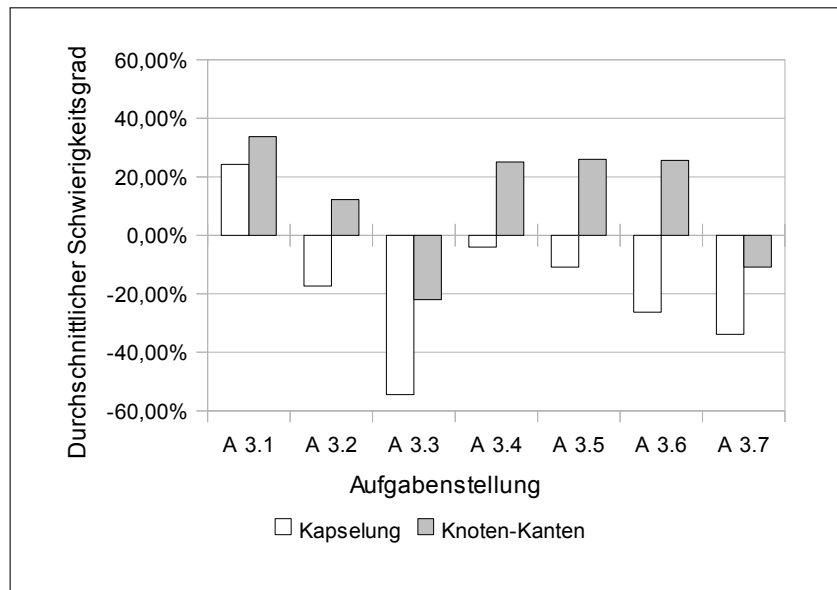


Abbildung 7.9: Durchschnittliche Bewertungen des Schwierigkeitsgrades je Aufgabenstellung in Aufgabenbearbeitung mit dem VIS. Positive Werte entsprechen einem geringen und negative Werte einem hohen Schwierigkeitsgrad.

Die **Bemessungen des Schwierigkeitsgrades** der einzelnen Aufgaben entspricht annäherungsweise der Fehlerrate. Allerdings existiert generell kein direkter Zusammenhang zwischen Fehlerrate und Schwierigkeitsgrad. Dies wird beispielsweise anhand der Auswertungen zu Aufgabe A 3.6 deutlich. Im Knoten-Kanten-Konzept wurde dieser Aufgabe zwar ein geringer Schwierigkeitsgrad zugeordnet, die Aufgabenbearbeitung enthielt aber auch die höchste Fehlerrate. Die Bewertungen der Probanden sind in Abbildung 7.9 zusammengefasst.

Die **Analyse des Ereignisprotokolls** für die Aufgabenbearbeitung ergibt weitere Auswertungskriterien und ermöglicht potenzielle Schlussfolgerungen über die Effektivität der Visualisierungskonzepte und des VIS. Im Rahmen der Auswertung dient hierfür primär die Bearbeitungszeit und die Anzahl der ausgeführten Aktionen je Proband als Vergleichsmerkmal und Messkriterium für Effizienz bzw. Effektivität. Die **Bearbeitungszeit** wird hierbei in zwei Gruppen gegliedert und umfasst die Zeiten im Übersichts- und Explorationsmodus. Abbildung 7.10 zeigt die durchschnittlichen Gesamtbearbeitungszeiten für das Kapselung-Konzept (11,5 Minuten) und das Knoten-Kanten-Konzept (9,38 Minuten). Die Visualisierungskonzepte weisen eine generell vergleichbare Bearbeitungszeit auf, wodurch Rückschlüsse auf höhere Effizienz bzw. Effektivität nur bedingt ermöglicht werden. Das Knoten-Kanten-Konzept besitzt hierbei jedoch eine minimal geringere Bearbeitungszeit, was möglicherweise auf die Vorkenntnisse zu klassischen Knoten-Kanten-Layouts zurückzuführen ist. Wie in Abbildung 7.11 ersichtlich, besitzen die durchschnittlichen Bearbeitungszeiten je Aufgabe, unabhängig von den Visualisierungskonzepten, die gleiche Tendenz. Das Kapselung-Konzept besitzt mehrheitlich eine geringfügig höhere Bearbeitungszeit, weist aber gleichzeitig auch stärkere Abweichungen auf. Die Aufgabe A 3.1 wurde im Kapselung-Konzept im Vergleich zum Knoten-Kanten-

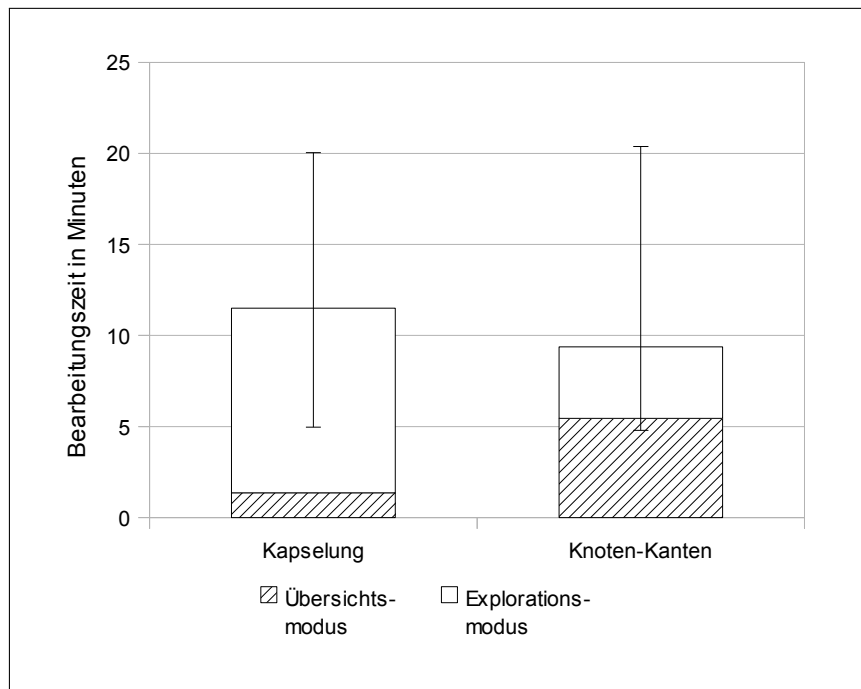


Abbildung 7.10: Durchschnittliche Gesamtbearbeitungszeit mit extremalen Abweichungen und anteiliger Gliederung in Übersichts- und Explorationsmodus

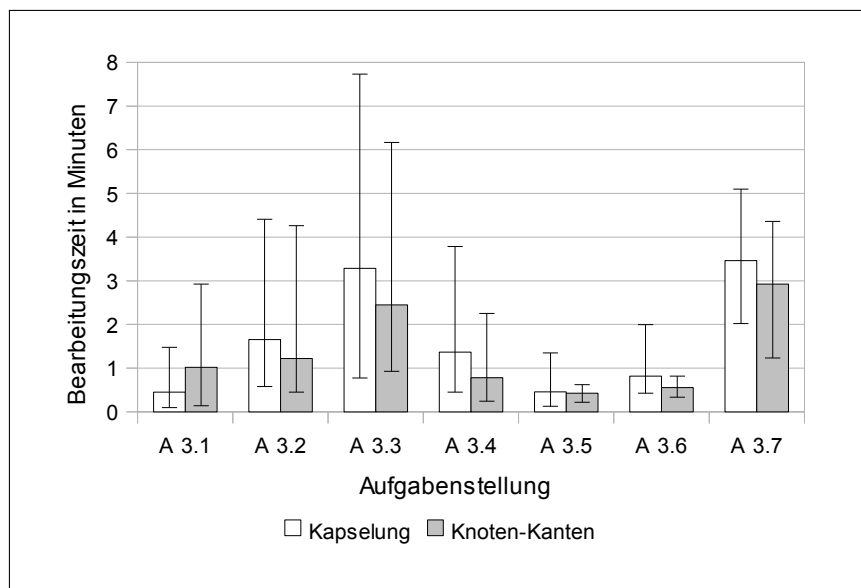


Abbildung 7.11: Durchschnittliche Aufgabebearbeitungszeiten mit extremalen Abweichungen

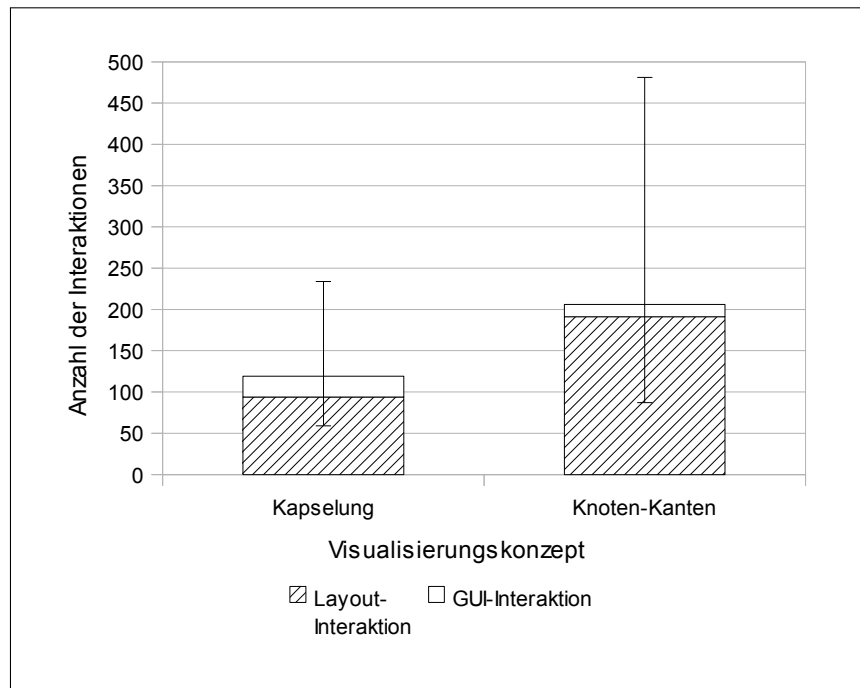


Abbildung 7.12: Durchschnittliche Anzahl der Interaktionen mit extremalen Abweichungen und anteiliger Gliederung in Layout- und GUI-Interaktionen

Konzept schneller bearbeitet. Die in dieser Aufgabe zu ermittelnde atomare Komponente konnte von den Probanden im Übersichtsmodus durch die effiziente Raumausnutzung des Kapselung-Layouts in kürzerer Zeit identifiziert werden.

Die protokollierten **Interaktionen** werden analog zur Bearbeitungszeit in zwei Gruppen gegliedert und beinhalten die Aktionen der Probanden in Bezug auf Layout- und GUI-Bestandteile. Zu den Interaktionen mit dem Layout zählen weiterhin Zoom- und Pan-Aktionen, Selektionen von Komponenten und das Öffnen und Verschieben der Informationsfenster. Die GUI-Interaktionen beziehen sich auf die Aktivierung der Schaltflächen. In Abbildung 7.12 sind die durchschnittlichen Anzahlen der Interaktionen je Visualisierungskonzept illustriert. Das Knoten-Kanten-Konzept weist eine annähernd doppelt so große Anzahl an Interaktionen aus, was primär auf die räumliche Konfiguration des Layouts und die damit verbundenen benötigten Zoom- und Pan-Aktionen zurückzuführen ist. Das Kapselung-Konzept beinhaltet hingegen einen größeren Anteil an GUI-Interaktionen. Dies ist vorrangig durch die notwendige Interaktion mit Gruppenkomponenten und die zugehörige Bedienung der Auswahlhistorien-Schaltfläche im Explorationsmodus bedingt. Weiterhin wurde im Kapselung-Konzept verstärkt die automatisierte Vergrößerungs- und Zentrierungsfunktion verwendet. Im Knoten-Kanten-Konzept wurden hingegen tendenziell manuelle Zoom- und Pan-Aktionen durchgeführt. Die berechneten Ergebnisse für Bearbeitungszeit und Interaktionen weisen allgemein **starke Abweichungen** auf, was unter anderem auf die unterschiedliche Vorgehensweise und die eingesetzten Interaktionsmethoden der Probanden zurückgeführt werden kann. Die Beobachtungen während der Benutzerstudie zeigten, dass die bereits angeführten Fehlerquellen weitere Ursa-

chen für relativ lange Bearbeitungszeiten und große Interaktionsanzahlen waren. Hatte ein Proband in Zusammenhang mit Fehlerquelle F 2 beispielsweise die Orientierung in der Struktur verloren, war er dazu gezwungen, den Ausgangszustand mittels der Reset-Schaltfläche herzustellen und die Informationsanalyse erneut zu beginnen.

Die Auswertung der **Einschätzungen und Bewertungen der Probanden** über die Aufgabenbearbeitung in der Benutzerstudie und den Umgang mit dem VIS sind in Abbildung 7.13 zusammengefasst. Die durchschnittlichen Bewertungen der Probanden zeigen eine einheitlich positive Resonanz. Keine der vordefinierten Aussagen wurde mehrheitlich negativ bewertet, was das grundsätzliche Potential des VIS und dessen Bestandteile verdeutlicht. Die Bewertungen der einzelnen Aussagen ermöglichen weitere Schlussfolgerungen auf die allgemeine Angemessenheit und Gebrauchstauglichkeit des VIS und die Eignung bezüglich der Zielgruppe und den Einsatzfeldern. In E 1–7 wurden die Visualisierungslayouts beispielsweise generell als verständlich und für die Abbildung hierarchischer Strukturen als geeignet empfunden. In E 12 erhielten die variablen Kantenlängen im Knoten-Kanten-Konzept zwar tendenziell positive Bewertungen, wurden aber, wie bereits angeführt, bei Abzähloperationen gleichzeitig als störend empfunden. Den Beobachtungen entsprechend, wurde die halbtransparente Navigationsvorschau in E 16 für das Kapselung-Konzept sowohl positiv, als auch negativ bewertet. In E 20 wurde die im Kapselung-Konzept auf den Explorationsmodus beschränkte Darstellung und Interaktion mit Gruppenkomponenten hingegen nicht als nachteilig eingestuft. Für eine Beurteilung der Angemessenheit der entwickelten Visualisierungskonzepte, ist zudem die Bewertung der Aussage E 8 von besonderem Interesse. Mit der Bewertung dieser Aussage brachten die Probanden ihre Bevorzugung bezüglich Indent-Layout oder Kapselung- bzw. Knoten-Kanten-Layout zum Ausdruck. Wie in Abbildung 7.14 gezeigt, wird das Kapselung-Konzept von 66,67% und das Knoten-Kanten-Konzept von 50% der Probanden bevorzugt.

In der letzten Phase der Benutzerstudie wurden Daten zu freien **Aussagen der Probanden** erhoben. Die freien Äußerungen bestätigten die allgemein positive Resonanz und die intuitive Bedienung des VIS wurde mehrfach thematisiert. Die Probanden des Kapselung-Konzepts bemerkten ferner die kompakte Darstellungsart und die Form, Fläche und Ikone der Komponenten als besonders hilfreich für die Orientierung in der Struktur. Als negativ wurden hingegen explizit die Fehlerquellen F 2 und F 3 eingeschätzt und die Farbkodierung des Navigationspfades konnte von einigen Probanden aufgrund von Rot-Grün-Sehschwäche nicht eindeutig erkannt werden. Im Zusammenhang mit dem Knoten-Kanten-Konzept wurde außerdem die Größe der Ikone bei Verkleinerung des Layouts als zu gering empfunden. Die entsprechenden Optimierungsvorschläge der Probanden wurden aufgenommen und etwaige Erweiterungen ausgearbeitet. Beispielsweise wurden die Intensivierung der Farbkodierung und die Hervorhebung der Strukturtiefe von Komponenten als Verbesserungsaspekte vorgeschlagen. Weiterhin wurden Vorschläge zur Erhöhung der Bildauflösung der Ikons, die Schattierung der Komponenten (Cushions), und die Abgrenzung bzw. Betonung des aktuellen Arbeitsbereichs

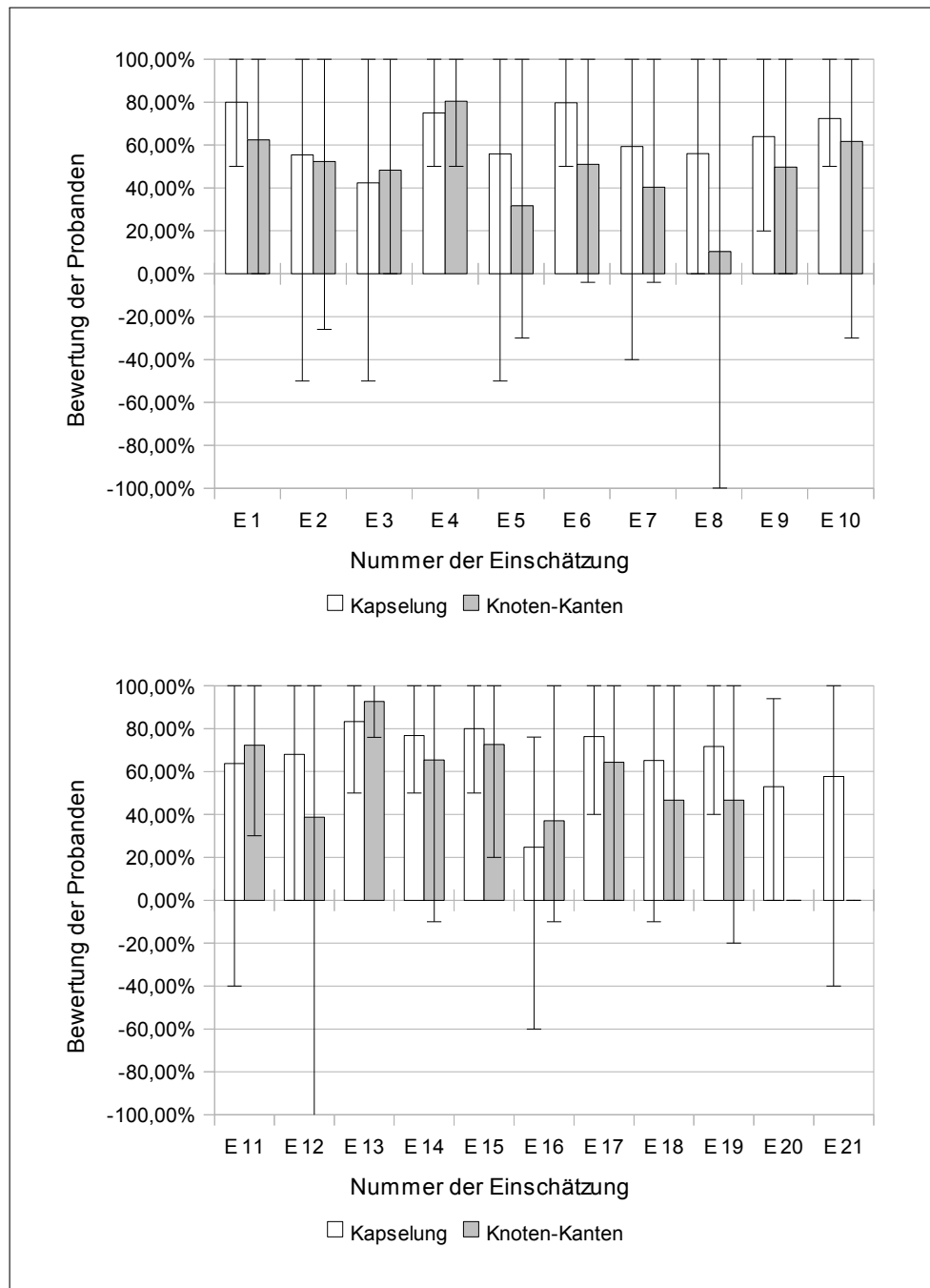


Abbildung 7.13: Durchschnittliche Bewertungen der vordefinierten Aussagen zum VIS mit extremalen Abweichungen. Positive Werte entsprechen einer Zustimmung und negative Werte einer Verneinung.

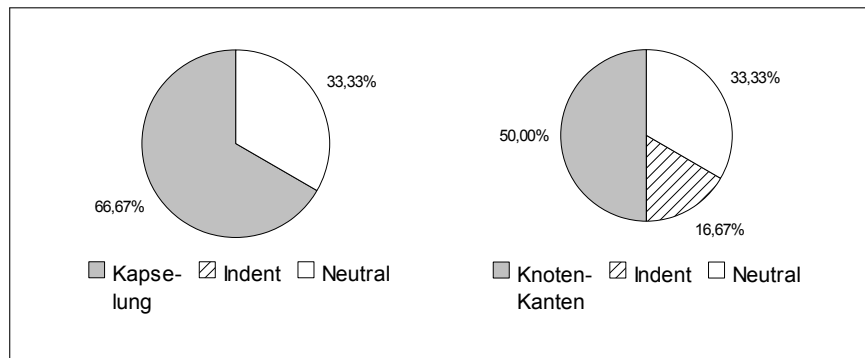


Abbildung 7.14: Anteilige Favorisierung der Probanden bezüglich der Layouttypen

unterbreitet. Unterschiedliche Größen und Flächen der Komponenten im Knoten-Kanten-Konzept wurden zudem befürwortet. Ein Proband thematisierte die Automatisierung der Informationsfenster und deren Anpassung an die Strukturnavigation zur Vermeidung von Überlagerungen mit dem Arbeitsbereich. In Bezug auf die Persistenz der Visualisierungslayouts und potenzielle Nutzung in kollaborativen oder verteilten Arbeitsprozessen, wurde insbesondere die Konstanz und Invarianz der Strukturabbildungen diskutiert. In diesem Zusammenhang ist es beispielsweise erforderlich, dass die Layoutkomponenten einer Basisstruktur auch bei Änderungen in Teilstrukturen stets eine einheitliche räumliche Konfiguration aufweisen. Die protokollierten Optimierungsansätze und die Diskussion über mögliche Erweiterungen, wie Suchfunktionen, weitere Navigationswerkzeuge und Widgets⁴⁹ bieten eine Grundlage für die Überarbeitung und potenzielle Erweiterung des VIS.

7.4 Diskussion und weiterführende Betrachtung

Die allgemeinen Beobachtungen und die Auswertung der spezifischen Ergebnisdaten zeigten, dass die wesentlichen Anforderungen an das VIS grundsätzlich erfüllt wurden. Weiterhin konnten die in Kapitel 7.1 aufgestellten Hypothesen zur Benutzerstudie im Wesentlichen bestätigt werden. Der These H 1 entsprechend weist insbesondere die Qualität der Ergebniserzeugung, neben dem grundlegenden Verständnis der Visualisierungslayouts, eine wesentliche Abhängigkeit zu den eingesetzten Interaktionsmethoden auf. Dieser Zusammenhang kam folglich in den gewonnenen Ergebnisdaten anhand von Abweichungen zum Ausdruck und spiegelt das Vorgehen der Probanden bei der Aufgabenbearbeitung wieder. In der weiteren Betrachtung sind die Ergebnisdaten darüber hinaus von den spezifischen Aufgabenstellungen abhängig. Eine bestimmte Aufgabenstellung könnte beispielsweise die Ermittlung einer speziellen Gruppenkomponente ohne Angabe der strukturellen Lage, wie den hierarchischen Zusammenhängen bis zu einem bestimmten Ausgangspunkt, fordern. Im Kapselung-Konzept ist der Benutzer folglich

⁴⁹ Ein Widget ist typischerweise eine visuelle Komponente einer grafischen Benutzungsoberfläche und dient der Darstellung von zusätzlichen oder unabhängigen Informationen sowie der zugehörigen Interaktion in einem abgegrenzten Bereich.

im ungünstigsten Fall (Worst Case Scenario) dazu gezwungen, große Teile des Layout im Explorationsmodus zu durchsuchen. In diesem Zusammenhang wäre das Knoten-Kanten-Konzept tendenziell im Vorteil, da die Gruppenkomponenten direkt abgebildet und auswählbar sind.

Eine Auswahl der Optimierungsansätze und aufgetretenen Fehlerquellen wurde bereits parallel in der fortführenden Entwicklung des Demonstrators adressiert. Durch die Erweiterung der Benutzungsschnittelle und den Einbezug andersartiger Bedientechniken wurden neue Randbedingungen geschaffen. Bei der Entwicklung und im Rahmen einer Präsentation des Demonstrators wurde das Kapselung-Konzept als besonders geeignet für die Interaktion mit Multi-Touch-Gesten identifiziert. Dies resultiert vorrangig aus der Form, Fläche und Größe der Layoutkomponenten, welche direkt als effektive Bedienelemente genutzt werden können. Um die große Interaktionsanzahl im Zusammenhang mit dem Knoten-Kanten-Konzept zu verringern, wurden ebenso erste experimentelle Interaktionskonzepte, wie beispielsweise ein Hyperbolic Tree Browser [LRP95], prototypisch umgesetzt.

Die Erweiterung des VIS unter Einbeziehung der gewonnenen Studienerkenntnisse erfordert weitere begleitende Benutzerstudien und fortführende Forschungsarbeit. Bestandteil dieser Studien können spezifische formulierte Aufgabenstellungen und Ausgangsstrukturen mit variabler Komplexität sein. Ein weiterer Studienaspekt ist der direkte Vergleich mit klassischen Indent-Konzepten und die Bewertung der Effektivität unter verschiedenen Voraussetzungen. Die visuellen Informationen sind dabei stets im Zusammenhang mit der Interaktion, den grundlegenden Voraussetzungen und den Eignungskriterien zu analysieren, um eine fortwährende Softwarequalitätssicherung zu gewährleisten.

8 Zusammenfassung und Ausblick

Die primäre Zielstellung dieser Arbeit war die Analyse und prototypische Umsetzung von Visualisierungs- und Interaktionskonzepten zur Steigerung der Effektivität in Arbeitsprozessen mit Produktstrukturen. Die Motivation resultierte aus der Erkenntnis, dass herkömmliche Präsentationstechniken von Produktstrukturen hohe Anforderungen an den Produktentwickler stellen und somit eine Schwachstelle im Produktentwicklungsprozess darstellen. Insbesondere die fortwährend steigende Komplexität in der Produktentwicklung erfordert Maßnahmen und geeignete Werkzeuge zur Beherrschung der Datenflut und Unterstützung bei der Informationsgewinnung. Bei der Konzeption und Entwicklung zweckmäßiger Methoden sind in diesem Zusammenhang vor allem die spezifischen Voraussetzungen und die Anpassungen an die Anforderungen der Zielgruppe und an die Einsatzfelder zu berücksichtigen. Ein wichtiger Aspekt ist die Kombination von abstrakten Produktstrukturen und 3D-Geometriemodellen, um die visuellen Wahrnehmungsfähigkeiten der Produktentwickler in Arbeitsprozessen zu fördern und etwaigen Belastungen, Fehlerquellen und Kommunikationsproblemen entgegen zu wirken.

Zur Gewährleistung einer zweckmäßigen Vorgehensweise wurden in der ersten Phase die Grundlagen der elementaren Ausprägungen von Produktstrukturen und ebenso die systematischen Prinzipien der Visualisierung und Interaktion analysiert. Weiterhin wurde in Bezug zum Stand der Forschung eine generelle Einteilung von Visualisierungstechniken vorgenommen und das Potenzial von neuartigen Präsentationsmedien und natürlichen Benutzungsschnittstellen ausgewertet. Der Stand der Visualisierung und Interaktion von Produktstrukturen innerhalb von erzeugenden und prozessbegleitenden Anwendungssystemen wurde in einem zusammenfassenden Überblick dargestellt. Aus den grundlegenden Anforderungen an ein Visualisierungssystem und den Erfordernissen der Produktentwicklung wurde folglich der konkrete Handlungsbedarf für die Arbeit abgeleitet.

Die Konzeption eines Visualisierungs- und Interaktionssystems (VIS) konkretisierte die Auswahl geeigneter Techniken für die Strukturabbildung. Die ausgewählten Konzepte wurden im Rahmen der systemtechnischen Realisierung prototypisch umgesetzt und in ein Basissystem eingebunden. Die visuelle Darstellung des Produktstammbaums stand hierbei im Fokus und die Konzepte wurden demnach hinsichtlich der Produktstrukturen erweitert und neuartig angewendet. Neben der Berücksichtigung von Integrationsaspekten und der Anbindung an vorherrschende Produktmanagementsysteme, wurde darüber hinaus eine experimentelle Erweiterung mittels Multi-Touch Technologie geschaffen. Die innovativen Präsentations- und Interaktionsmethodiken ermöglichen den Einbezug weiterer Anwendungskontexte und die Adaption für kollaborative Arbeitsprozesse.

Durch eine Benutzerstudie wurde die grundsätzliche Eignung und Gebrauchstauglichkeit des entwickelten Visualisierungs- und Interaktionssystems nachgewiesen. Die spezifischen Eigenschaften der Visualisierungskonzepte wurden gegenübergestellt und Grundlagen für fortführende Forschungsarbeiten konnten definiert werden. Die Auswertung verdeutlichte, dass die Reduktion der Defizite ordinärer Produktstrukturpräsentationen, die Handhabung komplexer Datenmengen und folglich die Erhöhung der Effektivität in der Produktentwicklung mittels der umgesetzten Konzepte potenziell möglich ist.

Auf der Basis der gewonnenen Erkenntnisse und der modularen Systemarchitektur der entstandenen Prototypen kann die systemtechnische Realisierung mit kalkulierbarem Aufwand ausgebaut und an weitere Anforderungen angepasst werden. Ein möglicher Erweiterungsansatz ist die Implementierung von Darstellungskonzepten für die verbleibenden elementaren Ausprägungen der Produktstrukturen. Die Ergänzung des Funktionsumfangs mit speziellen Werkzeugen für Entwurf und Bearbeitung der Strukturen und mit zusätzlichen Visualisierungskonzepten und Konzeptkombinationen ist denkbar. Um die Informationssuche der Benutzer zu unterstützen, könnten die Darstellungskonzepte mittels Fokus- und Detail-In-Kontext-Techniken verbessert werden. In einer aktuellen Weiterführung der Arbeit werden die protokollierten Kriterien der Probanden adressiert und die Generierungsperformance optimiert. Weitere Bestandteile sind dabei die vollständige Überführung in ein Mehrbenutzersystem und die Erforschung zugehöriger Prozess- und Interaktionsaspekte. In diesem Zusammenhang sind erneute Benutzerstudien durchzuführen, um Konzeption und Umsetzung zu validieren.

Die Funktionalität des VIS kann grundsätzlich als eigenständiges Visualisierungs- und Interaktionssystem für Produktstrukturen eingesetzt werden. Um die Anwenderakzeptanz zu fördern, ist jedoch je nach Anwendungskontext die stärkere Integration in bestehende Produktmanagementsysteme anzustreben. Gleichzeitig kann durch die Adaption bestimmter Komponenten der Betrieb auf mobilen Geräten oder als Online-Service mit Bezug auf die verteilte Produktentwicklung erreicht werden. Die Ausweitung des Einsatzgebietes auf benachbarte Phasen der Produktentwicklung oder unabhängige Bereiche bietet neue Herausforderungen an die Visualisierungs- und Interaktionstechniken. Vor allem Bereiche im Umfeld komplexer Daten- und Informationsbestände sind hierbei adressierbar. In Abbildung B.1 des Anhangs wurde der in der Arbeit verfolgte Ansatz exemplarisch auf die komplexe Struktur eines Softwaresystem generalisiert. Der Einsatz in anderen Anwendungskontexten erfordert erneute spezifische Anpassungen und Integrationen und ist Bestandteil weiterer interdisziplinärer Forschungsarbeiten.

Literaturverzeichnis

- [BB95] BAKER, M. P. ; BUSHELL, Colleen: After the Storm: Considerations for Information Visualization. In: *IEEE Computer Graphics and Applications* 15 (1995), S. 12–15. – ISSN 0272–1716
- [BBD06] BOTTGER, Joachim ; BALZER, Michael ; DEUSSEN, Oliver: Complex Logarithmic Views for Small Details in Large Contexts. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), September, S. 845–852. – ISSN 1077–2626
- [BD05] BALZER, Michael ; DEUSSEN, Oliver: Voronoi Treemaps. In: *Proceedings of IEEE Symposium on Information Visualization 2005*, 2005, S. 49–56
- [BD07] BALZER, M. ; DEUSSEN, O.: Level-of-Detail Visualization of Clustered Graph Layouts. In: *Asia-Pacific Symposium on Visualization 0* (2007), S. 133–140. ISBN 1–4244–0808–3
- [BDL05] BALZER, Michael ; DEUSSEN, Oliver ; LEWERENTZ, Claus: Voronoi treemaps for the visualization of software metrics. In: *Proceedings of the 2005 ACM Symposium on Software visualization*. New York, NY, USA : ACM, 2005 (SoftVis '05), S. 165–172
- [BHO10] BOSTOCK, Mike ; HEER, Jeff ; OGIEVETSKY, Vadim: *Protovis: A graphical toolkit for visualization*. 2010. URL <http://vis.stanford.edu/protovis/>. – [Online; Stand 28. Mai 2010]
- [BHW00] BRULS, M. ; HUIZING, K. ; WIJK, J. van: Squarified Treemaps. In: *Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization (TCVG 2000)*, IEEE Press, 2000, S. 33–42
- [BSW02] BEDERSON, Benjamin B. ; SHNEIDERMAN, Ben ; WATTENBERG, Martin: Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. In: *ACM Transactions on Graphics (TOG)* 21 (2002), October, S. 833–854
- [CC07] COLLINS, Christopher ; CARPENDALE, Sheelagh: VisLink: Revealing Relationships Amongst Visualizations. In: *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), S. 1192–1199. – ISSN 1077–2626
- [Che76] CHEN, Peter P.: The Entity-Relationship Model: Toward a Unified View of Data. In: *ACM Transactions on Database Systems* 1 (1976), S. 9–36

- [Che04] CHEN, Chaomei: *Information Visualization, Second Edition: Beyond the Horizon*. 2. Springer, 2004. – ISBN 1852337893
- [Coc04] COCKBURN, Andy: Revisiting 2D vs 3D implications on spatial memory. In: *Proceedings of the fifth conference on Australasian user interface - Volume 28*. Darlinghurst, Australia, Australia : Australian Computer Society, Inc., 2004 (AUIC '04), S. 25–31
- [DI10] DITTRICH, Elisabeth ; ISRAEL, Habakuk: *Informationsvisualisierung von Produktdaten in immersiver virtueller Realität*. 2010
- [DIN77] Norm DIN 199-2 1977. *CAD-Modelle, Zeichnungen und Stücklisten: Begriffe für Stücklisten und das Stücklistenwesen*
- [DIN98] Norm DIN EN ISO 9241-11 01 1998. *Ergonomie der Mensch-System-Interaktion - Teil 11: Anforderungen an die Gebrauchstauglichkeit*
- [DIN02] Norm DIN 199-1 2002. *CAD-Modelle, Zeichnungen und Stücklisten: Begriffe*
- [DIN03] Norm DIN 6789 2003. *Dokumentationssystematik: Produktdokumentation*
- [DIN08] Norm DIN EN ISO 9241-110 09 2008. *Ergonomie der Mensch-System-Interaktion - Teil 110: Grundsätze der Dialoggestaltung*
- [ET08] ELMQVIST, Niklas ; TSIGAS, Philippos: A Taxonomy of 3D Occlusion Management for Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 14 (2008), September, S. 1095–1109. – ISSN 1077–2626
- [FB95] FURNAS, George W. ; BEDERSON, Benjamin B.: Space-scale diagrams: understanding multiscale interfaces. In: *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 1995. – ISBN 0–201–84705–1, S. 234–241
- [GSA09] GALL, J. ; STOLL, C. ; AGUIAR, E. de ; THEOBALT, C. ; ROSENHAHN, B. ; SEIDEL, H. P.: Motion capture using joint skeleton tracking and surface estimation. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 0 (2009), S. 1746–1753
- [HBP02] HORNBAEK, Kasper ; BEDERSON, Benjamin B. ; PLAISANT, Catherine: Navigation Patterns and Usability of Zoomable User Interfaces With and Without an Overview. In: *ACM Transactions on Computer-Human Interaction* 9 (2002), S. 362–389

- [HD02] HABERFELLNER, Reinhard ; DAENZER, Walter F.: *Systems engineering: Methodik und Praxis*. 11nd ed. Zürich : Verlag Industrielle Organisation, 2002
- [HK06] HAN, Jiawei ; KAMBER, Micheline: *Data Mining: Concepts and Techniques*. 2nd ed. Morgan Kaufmann, 2006. – ISBN 1558609016
- [HKL99] HOFF, Kenneth E. III ; KEYSER, John ; LIN, Ming ; MANOCHA, Dinesh ; CULVER, Tim: Fast computation of generalized Voronoi diagrams using graphics hardware. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 1999 (SIGGRAPH '99). – ISBN 0–201–48560–5, S. 277–286
- [Hol06] HOLTEN, Danny: Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), September, Nr. 5, S. 741–748. – ISSN 1077–2626
- [HTS09] HORN, Michael S. ; TOBIASZ, Matthew ; SHEN, Chia: Visualizing Biodiversity with Voronoi Treemaps. In: *Proceedings of the 2009 Sixth International Symposium on Voronoi Diagrams*. Washington, DC, USA : IEEE Computer Society, 2009 (ISVD '09). – ISBN 978–0–7695–3781–8, S. 265–270
- [Isr09] ISRAEL, J.H.: Hybride Interaktionstechniken des immersiven Skizzierens in frühen Phasen der Produktentwicklung. (2009)
- [JDG02] JU, Lili ; DU, Qiang ; GUNZBURGER, Max: Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations. In: *Parallel Computing* 28 (2002), October, S. 1477–1500. – ISSN 0167–8191
- [JF98] JUL, Susanne ; FURNAS, George W.: Critical zones in desert fog: aids to multiscale navigation. In: *Proceedings of the 11th annual ACM Symposium on User interface software and technology*. New York, NY, USA : ACM, 1998 (UIST '98). – ISBN 1–58113–034–1, S. 97–106
- [KB07] KALTENBRUNNER, Martin ; BENCINA, Ross: reacTIVision: a computer-vision framework for table-based tangible interaction. In: *TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction*. New York, NY, USA : ACM, 2007. – ISBN 978–1–59593–619–6, S. 69–74
- [KFG06] KRAUSE, Frank L. ; FRANKE, Hans J. ; GAUSEMEIER, Jürgen: *Innovationspotenziale in der Produktentwicklung*. 1. München : Hanser Fachbuchverlag, 2006. – 3–30 S. – ISBN 3446406670

- [KFKW10] KAMMER, Dietrich ; FREITAG, Georg ; KECK, Mandy ; WACKER, Markus: Taxonomy and Overview of Multi-touch Frameworks: Architecture, Scope and Features. In: *ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (2010)
- [Kü04] KÜHL, Nikolaus: *Produkt Daten Management*. 2004. – Vorlesungsskript Fachhochschule Frankfurt am Main
- [LA94] LEUNG, Y. K. ; APPERLEY, M. D.: A review and taxonomy of distortion-oriented presentation techniques. In: *ACM Transactions on Computer-Human-Interaction* 1 (1994), June, S. 126–160. – ISSN 1073–0516
- [LBS85] LEE, SK ; BUXTON, William ; SMITH, K. C.: A multi-touch three dimensional touch-sensitive tablet. In: *SIGCHI Bull.* 16 (1985), April, S. 21–25. – ISSN 0736–6906
- [Les01] LESZINSKI, Christoph M.: *Ein Visualisierungs- und Navigationsassistent für Produktstrukturen in der Produktentwicklung*. Bochum, Ruhr-Universität Bochum, Diss., 2001
- [LF08] LÜ, Hao ; FOGARTY, James: Cascaded treemaps: examining the visibility and stability of structure in treemaps. In: *Proceedings of graphics interface 2008*. Toronto, Ont., Canada, Canada : Canadian Information Processing Society, 2008 (GI '08). – ISBN 978–1–56881–423–0, S. 259–266
- [LRP95] LAMPING, John ; RAO, Ramana ; PIROLI, Peter: A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 1995 (CHI '95), S. 401–408
- [LRZ10] LAUFS, Uwe ; RUFF, Christopher ; ZIBUSCHKA, Jan: MT4j: A Cross-platform Multi-touch Development Framework. In: *ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (2010)
- [LVC03] LYMAN, Peter ; VARIAN, Hal R. ; CHARLES, Peter ; GOOD, Nathan ; JORDAN, Laheem L. ; PAL, Joyojeet: *How much information*. 2003. URL <http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/>
- [Maz09] MAZZA, Riccardo: *Introduction to Information Visualization*. London : Springer, 2009
- [MH98] MELANCON, Guy ; HERMAN, Ivan: Circular drawings of rooted trees. Ams-

- terdam, The Netherlands, The Netherlands : CWI (Centre for Mathematics and Computer Science), 1998. – Forschungsbericht
- [PHP03] PFITZNER, Darius ; HOBBS, Vaughan ; POWERS, David: A unified taxonomic framework for information visualization. In: *Proceedings of the Asia-Pacific Symposium on Information visualisation - Volume 24*. Darlinghurst, Australia, Australia : Australian Computer Society, Inc., 2003 (APVis '03). – ISBN 1–920682–03–1, S. 57–66
- [PXY05] PHAN, Doantam ; XIAO, Ling ; YEH, Ron ; HANRAHAN, Pat ; WINOGRAD, Terry: Flow Map Layout. In: *Proceedings of the IEEE Symposium on Information Visualization*, 2005, S. 219–224
- [RT07] RONG, Guodong ; TAN, Tiow-Seng: Variants of Jump Flooding Algorithm for Computing Discrete Voronoi Diagrams. In: *Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering*. Washington, DC, USA : IEEE Computer Society, 2007. – ISBN 0–7695–2869–4, S. 176–181
- [Saf08] SAFFER, Dan: *Designing Gestural Interfaces: Touchscreens and Interactive Devices*. O'Reilly Media, Inc., 2008. – ISBN 0596518390, 9780596518394
- [Sch10] SCHULZ, Hans-Jörg: *Explorative Graph Visualization*. Rostock, Universität Rostock, Diss., 2010
- [Sei08] SEIFFERT, Ulrich: *Virtuelle Produktentstehung für Fahrzeug und Antrieb im Kfz: Prozesse, Komponenten, Beispiele aus der Praxis*. Vieweg + Teubner, 2008 (ATZ-MTZ-Fachbuch). – ISBN 3834803456
- [SFL10] SUD, Avneesh ; FISHER, Danyel ; LEE, Huai-Ping: Fast Dynamic Voronoi Treemaps. In: *International Symposium on Voronoi Diagrams in Science and Engineering 0* (2010), S. 85–94
- [SGM05] SUD, Avneesh ; GOVINDARAJU, Naga ; MANOCHA, Dinesh: Interactive computation of discrete generalized voronoi diagrams using range culling. In: *In Proceedings International Symposium on Voronoi Diagrams in Science and Engineering*, 2005
- [Shi10] SHI, Nianyu: *Interaktive 3D-Visualisierung von komplexen Datenstrukturen mechatronischer Produkte*. Berlin, Technische Universität Berlin, Diss., 2010
- [Shn91] SHNEIDERMAN, Ben: Tree visualization with Tree-maps: A 2-d space-filling approach. In: *ACM Transactions on Graphics* 11 (1991), S. 92–99

- [Shn96] SHNEIDERMAN, Ben: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: *IEEE Visual Languages*. College Park, Maryland 20742, U.S.A., 1996, S. 336–343
- [Sie10] *Open product lifecycle data sharing using XML*. White paper, Siemens Product Lifecycle Management Software Inc., Juli 2010
- [SJ07] SEARS, Andrew (Hrsg.) ; JACKO, Julie A. (Hrsg.): *The Human Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. 2nd ed. Mahwah, NJ : CRC Press, 2007. – ISBN 0805858709
- [SM00] SCHUMANN, Heidrun ; MÜLLER, Wolfgang: *Visualisierung: Grundlagen und allgemeine Methoden*. 1st ed. Rostock : Springer Berlin, 2000. – ISBN 3540649441
- [SS77] SMITH, John M. ; SMITH, Diane C. P.: Database abstractions: aggregation and generalization. In: *ACM Transactions on Database Systems* 2 (1977), June, S. 105–133. – ISSN 0362–5915
- [TCM06] TARINI, Marco ; CIGNONI, Paolo ; MONTANI, Claudio: Ambient Occlusion and Edge Cueing for Enhancing Real Time Molecular Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), Nr. 5, 1237–1244. URL <http://qutemol.sourceforge.net>. – ISSN 1077–2626
- [TDBET98] TOLLIS, Ioannis G. ; DI BATTISTA, Giuseppe ; EADES, Peter ; TAMASSIA, Roberto: *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1998
- [TS07] TU, Ying ; SHEN, Han-Wei: Visualizing Changes of Hierarchical Data using Treemaps. In: *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), November, S. 1286–1293
- [Tuf01] TUFTE, Edward R.: *The Visual Display of Quantitative Information*. 2nd ed. Graphics Press, 2001. – ISBN 0961392142
- [VDI03] Norm VDI 3633 2003. *Simulation von Logistik-, Materialfluss- und Produktionssystemen: Simulation und Visualisierung*
- [Wal05] *Walrus: Graph Visualization Tool*. 2005. URL <http://www.caida.org/tools/visualization/walrus/>
- [War04] WARE, Colin: *Information Visualization, Second Edition: Perception for Design*. 2. Morgan Kaufmann, 2004. – ISBN 1558608192

- [Wik10] WIKIPEDIA: *Struktur*. 2010. URL <http://de.wikipedia.org/wiki/Struktur>. – [Online; Stand 20. Oktober 2010]
- [YKSJ07] YI, Ji S. ; KANG, Youn a. ; STASKO, John ; JACKO, Julie: Toward a Deeper Understanding of the Role of Interaction in Information Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), November, S. 1224–1231. – ISSN 1077–2626

Anhang A: Exemplarischer Fragebogen der Benutzerstudie

Datum: **10.11.2010**

Uhrzeit: **18:14**

Teilnehmer-ID: **9**

Visualisierungstyp (Kapselung, Knoten-Kanten): **Kapselung**

Evaluation

Datenschutzerklärung

Alle Daten werden freiwillig zur Verfügung gestellt und finden ausschließlich im Rahmen der Evaluation Verwendung. Darüber hinaus werden die Daten nicht an Dritte weitergegeben oder intern für andere Zwecke genutzt. Alle persönlichen Angaben werden absolut vertraulich behandelt und dienen lediglich der Zuordnung bzw. Kategorisierung der Evaluationsergebnisse.

Allgemeine Fragen

Alter	28	
Beruf bzw. Studiengang und Semester	Wissenschaftlicher Mitarbeiter	
Haben Sie Vorkenntnisse im Umgang mit hierarchischen Strukturen (z.B. Organisation von Dateien in Ordnern)	<input checked="" type="checkbox"/> Ja	Nein
Haben Sie Vorkenntnisse im CAD Umfeld	<input checked="" type="checkbox"/> Ja	Nein
Haben Sie Vorkenntnisse im Umgang mit virtuellen Baugruppen	<input checked="" type="checkbox"/> Ja	Nein

Aufgabenbeschreibung

- Dem Proband werden verschiedene Strukturaufgaben gestellt, welche mit Hilfe des zur Verfügung gestellten Materials zu lösen sind.
- Der Proband hat die Möglichkeit sich die Aufgabenstellung jeweils im Vorab durchzulesen und gegebenenfalls Verständnisfragen zu klären.
- Die Aufgaben sind möglichst schnell zu absolvieren und die Vollendung ist dem Versuchsbetreuer durch Ansagen mitzuteilen.
- Je nach Aufgabenstellung sind die Ergebnisse dem Versuchsbetreuer mitzuteilen oder schriftlich in den dafür vorgesehenen Feldern (siehe unten) festzuhalten.
- Falls eine Aufgabe nicht korrekt gelöst wurde, wird der Versuchsbetreuer darauf hinweisen und die Aufgabe ist weiter zu bearbeiten.
- Weitere Hilfestellungen zur Aufgabenbearbeitung sind nicht zugelassen.

Strukturtest 1

Erläuterungen von Terminologie und Aufgabenstellung am Beispiel Struktur-Nr. 0.
Folgende Aussagen werden dem Probanden erklärt:

- Knoten 1 ist der Wurzelknoten und ein Gruppenknoten
- Knoten 7 hat den Geschwisterknoten 6 und ist ein atomarer Knoten
- Knoten 10 hat die Elternknoten 5 und 1 und ist ein Gruppenknoten
- Knoten 5 hat die direkten Kind- bzw. Subknoten 9 und 10 sowie unter anderem Knoten 12 als indirekten Kind- bzw. Subknoten
- Die Strukturtiefe 3 (Wurzelknoten hat ST 0) haben die Knoten 11, 12, 13, 14

1. Aufgabenstellung

- 1.1. Was ist/sind der/die Gruppenknoten.
- 1.2. Welche(r) Knote(n) hat/haben die meisten direkten Kind- bzw. Subknoten.
- 1.3. Was ist/sind der/die Elternknoten von Knoten 94.
- 1.4. Was ist/sind der/die Geschwisterknoten von Knoten 21.
- 1.5. Welche(r) Knote(n) hat/haben die Strukturtiefe 2.

= 28 sec

Strukturtest 2

2. Aufgabenstellung

- 2.1. Die vorliegende Abbildung zeigt eine Layoutdarstellung analog zur vorgestellten Anwendung. Diese Darstellung ist in ein Indent-Layout entsprechend Strukturtest 1 zu überführt.
 - Zum Zeichnen bitte den Platz unterhalb der Aufgabenstellung oder auf der Rückseite des Fragebogens nutzen

= 56 sec

Anwendungstest

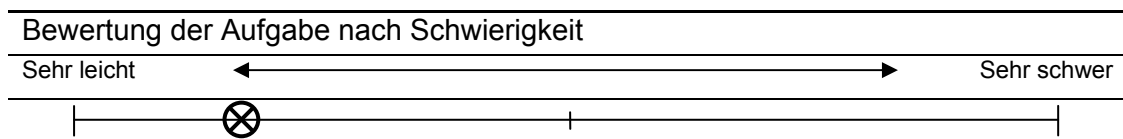
Ausgangssituation und Voreinstellung der Prototypen für die Aufgabenbearbeitung

- Applikation befindet sich im Explorationsmodus
- Keine Knoten sind aktiv oder ausgewählt
- Zentrierung des Graphen und Zoom auf Wurzelknoten

3. Aufgabenstellung

3.1. Finden und Selektieren Sie die vorliegende Komponente.

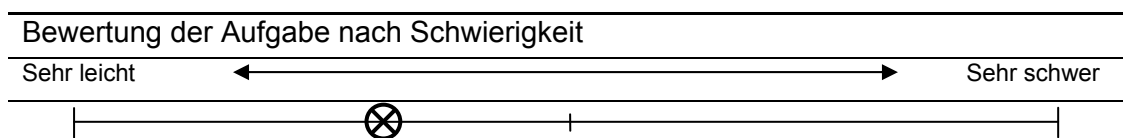
- Vollendung und Ergebnis bitte ansagen!
- Anzahl der Hinweise (Versuchsbetreuer): **0**



3.2. Notieren Sie alle Elternkomponenten der fokussierten Komponente bis zur Wurzelkomponente.

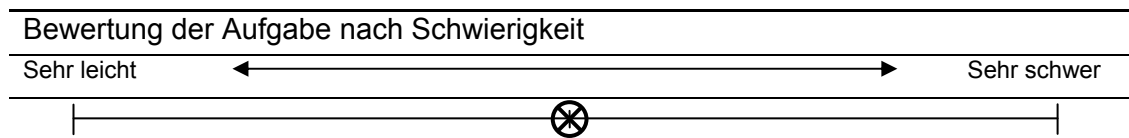
- Besondere Ausgangssituation:
 - Zoom auf festgelegte Komponente
 - Die farbliche Markierung (Navigationspfad) von ausgewählter Komponente bis zur Wurzelkomponente ist bei dieser Aufgabe deaktiviert

- Elternkomponenten (UIDs): **77, 70, 2, 1**
- Anzahl der Hinweise (Versuchsbetreuer): **0**



3.3. Welche der beiden vorliegenden Komponenten hat die meisten direkten und indirekten Subkomponenten bis zum Verzweigungsende (atomare Komponenten und Gruppenkomponenten).

- Komponente mit meisten Subkomponenten (A, B): **B**
- Anzahl der Hinweise (Versuchsbetreuer): **0**



3.4. Welche der beiden vorliegenden Komponenten hat die meisten direkten und indirekten Subkomponenten bis zum Verzweigungsende (atomare Komponenten und Gruppenkomponenten). Notieren Sie die Anzahl der Subkomponenten.

- Komponente mit meisten Subkomponenten (A, B) und Anzahl: **B (3)**
- Anzahl der Hinweise (Versuchsbetreuer): **0**



3.5. Wie oft kommt die vorliegende Komponente in der Gesamtstruktur vor. Identifikation durch Namen, bildliche Darstellung und Component-ID. Die UUIDs sind zu vernachlässigen!

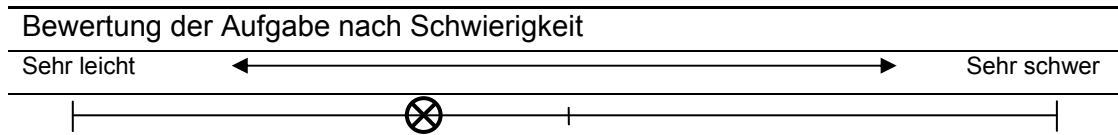
- Anzahl: **4**
- Anzahl der Hinweise (Versuchsbetreuer): **0**



3.6. Wie viele direkte Subkomponenten hat die vorliegende Komponente (atomare Komponenten und Gruppenkomponenten).

➤ Anzahl: **9**

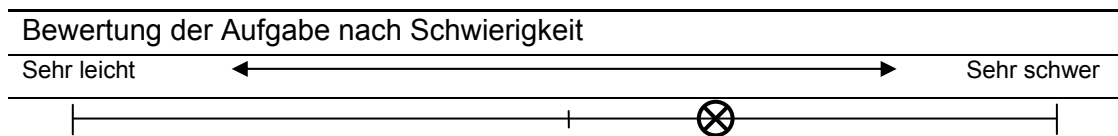
➤ Anzahl der Hinweise (Versuchsbetreuer): **0**



3.7. Es werden Ihnen zwei Prototypen zur Verfügung gestellt, welche unabhängig voneinander zu bedienen sind. Die beiden Layoutdarstellung der Prototypen haben eine unterschiedliche Struktur. Vergleichen Sie die beiden Layouts und finden Sie alle strukturellen Unterschiede, wobei die räumliche Konfiguration zu vernachlässigen ist. Notieren Sie die Differenz als Ergebnis.

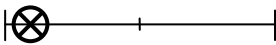
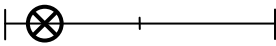
➤ Unterschiede: **Komponente „Slider Rubber“ zu viel (rechts)**

➤ Anzahl der Hinweise (Versuchsbetreuer): **0**



Einschätzung und Bewertung der Evaluation und Aufgabenbearbeitung

		zutreffend	nicht zutreffend
1	Die Layoutdarstellung der Struktur ist verständlich und mit minimaler Einarbeitungszeit nachvollziehbar		
2	Die strukturellen Beziehungen in der Layoutdarstellung sind intuitiv verständlich und nachvollziehbar		
3	Im Verlauf der Aufgabenbearbeitung entsteht ein Lern- bzw. Erinnerungsprozess zur abgebildeten Struktur		
4	Falls 3. zutreffend - Die Strukturkenntnis ist im weiteren Verlauf der Aufgabenbearbeitung hilfreich und nachfolgende Aufgaben sind leichter zu lösen		
5	Für die Aufgabenbearbeitung sind keine weiteren Hilfestellungen nötig		
6	Die räumliche Konfiguration der Layoutdarstellung und die Komponentenpositionen sind verständlich		
7	Die Layoutdarstellung ermöglicht eine intuitive Strukturnavigation und einfache Orientierung selbst bei starker Vergrößerung des Layouts		
8	Die vorgestellte Layoutdarstellung wird gegenüber einem Indent-Layout bevorzugt		
9	Die Strukturdarstellung und die Farbkodierung sind ästhetisch ansprechend		
10	Die farbliche Kodierung ist für die Orientierung in der Struktur hilfreich und unterstützend		
11	Die bildliche Komponentendarstellung ist für die Identifikation und das Strukturverständnis hilfreich und unterstützend		
12	Die Größe der Zellen (Kapselung) bzw. Länge der Kanten (Knoten-Kanten) ist für Aufgabenbearbeitung und Strukturverständnis hilfreich und unterstützend		
13	Der Übersichtsmodus ist hilfreich und unterstützend		
14	Der Explorationsmodus ist hilfreich und unterstützend		
15	Die verfügbaren Interaktionsmethoden, wie Zoom, Panning, Autofokus, etc., sind ausreichend		
16	Die halbtransparente Vorschau der Subkomponenten bei der Komponenten-Aktivierung ist hilfreich und für die Interaktion unterstützend		
17	Der Autofokus und -zoom bei der Selektion von Komponenten ist hilfreich und für Interaktion und Orientierung unterstützend		
18	Die Informationsfenster sind für die Darstellung der Komponenten-Informationen geeignet		
19	Die Informationsfenster sind intuitiv zu bedienen		

20	<u>Nur Kapselung</u> : Die auf den Explorationsmodus beschränkte Darstellung und Selektion von Gruppenkomponenten ist nicht von Nachteil	
21	<u>Nur Kapselung</u> : Die adaptiven Randlinien der Zellen sind für das Strukturverständnis und die Identifikation von Geschwisterkomponenten hilfreich	

Persönliche Einschätzung und Meinung	<ul style="list-style-type: none"> • <i>Intuitive Bedienung</i> • <i>Übersichtsmodus war aufgrund der überschaubaren Anzahl von Komponenten sehr hilfreich</i>
--------------------------------------	--

Verbesserungs- und Optimierungsvorschläge Beispiele: <ul style="list-style-type: none"> • Interaktion: Rotation, Lupe, Auswahlwerkzeuge, etc. • Eingabe Geräte: Tastenkombinationen, etc. • Layoutdarstellung: Komponentenpositionen und Darstellung (2D, 3D), Farbkodierung, etc. • Widgets: Breadcrumb-Navigation, Übersichtsdarstellung des Layout, Kompass zum Wurzelknoten, etc. 	<ul style="list-style-type: none"> • <i>Strukturtiefe der Komponenten stärker hervorheben</i> • <i>Farbkodierung des Navigationspfads ist wegen Rot-Grün-Farbblindheit ungeeignet</i>
--	---

Anhang B: Exemplarische Generalisierung des VIS

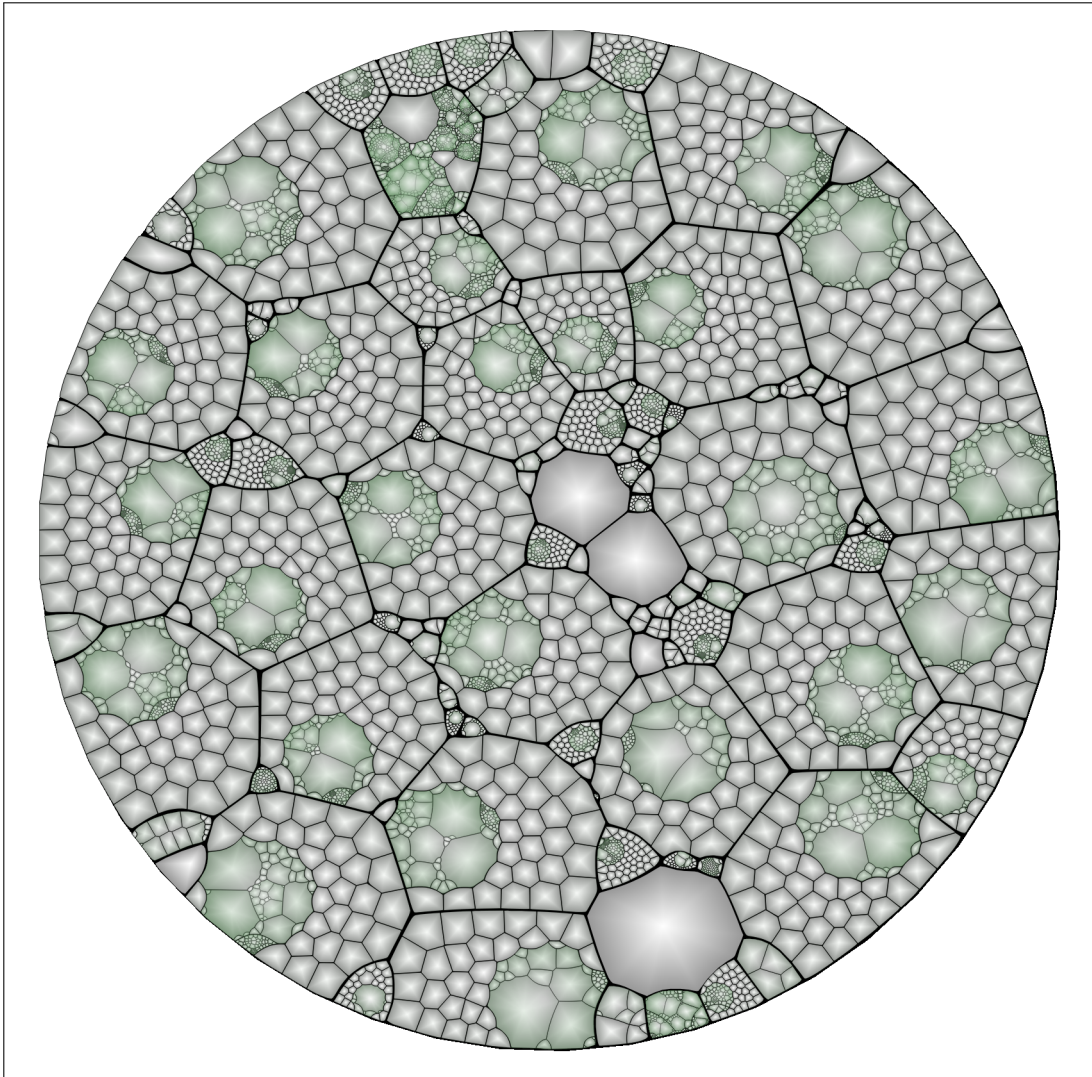


Abbildung B.1: Visualisierung eines komplexen Softwaresystems mit insgesamt 12434 Strukturelementen auf 9 Hierarchieebenen

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Berlin, 27. Januar 2011

A handwritten signature in black ink, reading "Martin Böhlig". The signature is written in a cursive style with a large, looped 'M' and a long, trailing 'g'.